AD-A043 864 OPERATIONS RESEARCH INC SILVER SPRING MD F/G 5/1
DEVELOPMENT OF A PRELIMINARY AUTOMATED TOTAL SYSTEMS MODEL FOR --ETC(U)
FEB 70 T N KYLE, R D HEILBRON, J D AVILA N00025-67-C-0031
UNCLASSIFIED ORI-TR-583-VOL-4 NL

1 OF 3
AD A043864

ORI

Operations Research, Inc. A Subsidiary of Leasco Systems Corporation

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Development of a Preliminary Automated Total Systems Model for the Integrated Facilities Requirements Study (IFRS) Phase II, Volume IV, Programmer's Manual	5. TYPE OF REPORT & PERIOD COVERED Final Report of 9 Feb '70 6. PERFORMING ORG. REPORT NUMBER ORI TR;NO.583, VOL IV of IV
7. AUTHOR(s) Thomas N. Kyle, Frederick L. McCoy, Richard D. Heilbron, Thomas L. Shaffer, John D. Avila, Dr. Joan L. Turek	8. CONTRACT OR GRANT NUMBER(s) N00025-67-C-0031 (NBY) 78672
9. PERFORMING ORGANIZATION NAME AND ADDRESS Operations Research, Inc. Silver Spring, MD	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS N.A.
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Facilities Engineering Command Department of the Navy Washington, DC	12. REPORT DATE 9 February 1970 13. NUMBER OF PAGES 257
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* SAME	15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Statement No. A - Distribution of this Document is unlimited; approved for public release

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

SAME

18. SUPPLEMENTARY NOTES

N.A.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Facilities	Fleet	Simulation	Static
Requirements	Training	Programming	Air
Dynamic	Aircraft	Management	Readiness
Optimization	Pilot	Planning	Model

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report documents the second phase of the multi-phase Integrated Facilities Requirements Study (IFRS).

In Phase I, two analytic submodels were developed. The first, a Logistics Support Requirements Generator, estimates personnel, aircraft, and fuel requirements for each phase of undergraduate pilot training at the Naval Air Training Command (NATRACOM). The second, a Pacing Facilities Requirements sub-

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

model, calculates facility requirements for each phase of training.

The purpose of the Phase II study was to develop a preliminary total systems IFRS management planning tool (including the two submodels developed in Phase I, as well as Base Loading, Facilities Excess/Deficiency, and Total Cost submodels), and automate the model so that it provides quick, accurate, and relevant information for use in the decision-making process. The present IRFS model is working to provide useful information to the decision-maker. Refinement and expansion of the present Phase II model will be completed in Phase III.

This report is composed of four volumes. Volume I contains a summary of the IFRS management planning tool. A detailed discussion of each of the five submodels and associated data files is contained in Volume II. A manual discussing the use of the automated model is provided in Volume III and the programmer's manual is contained in Volume IV.

OPERATIONS RESEARCH, Inc.

SILVER SPRING, MARYLAND

DEVELOPMENT OF A PRELIMINARY AUTOMATED TOTAL SYSTEMS MODEL FOR THE INTEGRATED FACILITIES REQUIREMENTS STUDY (IFRS) PHASE II

VOLUME IV—PROGRAMMER'S MANUAL

9 February 1970

Prepared under Contract N00025-67-C-0031
(NBy-78672) for the Naval Facilities Engineering Command
Department of the Navy
Washington, D.C.

FOREWORD

This report documents the second phase of the multi-phase Integrated Facilities Requirements Study (IFRS). It has been prepared for the Systems Analysis Division of the Office of the Assistant Commander for Facilities Planning (Code 20), Naval Facilities Engineering Command (NAVFAC), Department of the Navy, as part of Contract N00025-67-C-0031 (NBy-78672) awarded to Operations Research, Inc., in June 1969.

In Phase I, two analytic submodels were developed. The first, a Logistics Support Requirements Generator, estimates personnel, aircraft, and fuel requirements for each training phase. The second, a Pacing Facilities Requirements Submodel, calculates facility requirements for each phase of training.

The purpose of the Phase II study was to develop a preliminary total systems IFRS model (including the two submodels developed in Phase I, as well as base loading, facilities excess/deficiency, and total cost submodels), and automate the model so that it provides quick, accurate, and relevant information for use in the decision-making process. The present IFRS model is working to provide useful information to the decision maker. Refinement and expansion of the present Phase II model will be completed in Phase III.

This report is composed of four volumes. Volume I contains a summary of the IFRS management planning tool. A detailed discussion of each of the five submodels and associated data files is contained in Volume II. A manual discussing the use of the automated model is provided in Volume III and the programmer's manual is contained in Volume IV.

The IFRS model was developed and programmed by staff members of the Economic Analysis Division of Operations Research, Inc., under the direction of Dr. William J. Leininger, Vice President and Division Director, and Thomas N. Kyle, Project Manager. The project team members were Richard D. Heilbron, John H. Avila, Frederick L. McCoy, Thomas L. Shaffer, and Dr. Joan L. Turek.

Mr. Dennis Whang of the Systems Analysis Division of Facilities Planning was contract monitor for NAVFAC. In addition, valuable assistance was provided by many other Navy personnel including, in particular, those in the Office of the Staff Civil Engineer and the Training/Plans Division of the Naval Air Training Command and in the Systems Analysis Division of NAVFAC. The authors gratefully acknowledge the contributions made by all of these people to the development of the IFRS model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

1.1 This volume describes the overall system characteristics and flow for all computer programs included in the automated Integrated Facilities Requirements Study (IFRS). The purpose of the programmer's manual is to provide the verbal description, flow charts, variable dictionary, program and subroutine dictionary, and program listing for each of the computer programs that constitute the automated model of the IFRS. This programmer's manual provides CNATRA personnel with the information required to understand the logic of the IFRS programming and to make changes to the programs as necessary.

1.2 The IFRS programs have been written in FORTRAN IV for use on a General Electric (GE) 635, Mark II, time-sharing computer system. It is assumed that the programmer using this documentation is fully acquainted with G. E. Time-Sharing Mark II-FORTRAN.[1/]

[1/] Converting the computer programs contained in the automated IFRS model for use on other FORTRAN IV systems would require major revisions to each individual computer program's input and output, due to a special feature in the GE time-sharing FORTRAN that allows unformatted input and output, an adaptation of a BASIC language feature. The authors of the IFRS computer programs utilized this feature, when practical, to provide the user with maximum terminal input flexibility. In addition, storage restrictions might, in other FORTRAN IV systems, require resegmenting the IFRS system so maximum program storage requirements would not be violated.

1.3 Because of the "in core" word limitation imposed by the GE 635 time-sharing computer, the five submodels of the automated IFRS model are comprised of 15 different computer programs. This system will only permit a program of approximately 18,000 36-bit words to reside within the computer memory at any one time. Since the total IFRS model is much longer than this limit (approximately 80,000 36-bit words in the unsegmented version), it was necessary to use 15 operating programs. Table 1 lists each of these programs, their source and compiled names, and source and compiled lengths. The asteric in the sixth character "*" in the compiled name allows these programs to be accessed by all users (with certain restrictions) having similar GE user numbers.

1.4 Figure 1 shows the overall flow through the 15 programs in the IFRS model. Also shown are the various data files either read or written during the course of a run through all the computer programs. A brief description of all computer programs and data files utilized by the automated IFRS model is provided in Table 2.

ORGANIZATION OF MANUAL

1.5 The remaining portion of this manual describes each of the 15 computer programs. For each program, a detailed verbal description, flow charts, variable dictionary, routine dictionary (briefly describing the function of each main program and its subroutines), and program listing are provided. In addition, the program listings for the four utility programs are included.

1.6 Wherever possible, variable names were selected as mnemonics. For example, in PROGRAM LSRM, the variable LEVLSR denotes the level of complexity within the LSR Generator. In PROGRAM PART2, the variable ACREQ denotes the aircraft required, etc.

TABLE 1

AUTOMATED IFRS PROGRAMS

Normal Running Sequence	Source Program Name	Compiled Program Name	Source Program Length (1,000 words)	Compiled Program Length (1,000 words)
1	LSRM	XLSRM*	2.6	2.6
2	LSR1	XLSR1*	22.3	24.7
3	LSR2	XLSR2*	22.1	22.1
4	LSR3	XLSR3*	11.7	14.3
5	LSR4	XLSR4*	7.3	8.4
6	PART2	PART2*	3.6	5.2
7	PART3	PART3*	13.0	18.2
8	PART4	PART4*	11.7	14.3
9	PARTY	PARTY*	9.7	10.4
10	PART5	PART5*	9.1	9.1
11	PARTX	PARTX*	3.9	6.5
12	PART6	PART6*	11.7	15.6
13	PART7	PART7*	14.3	14.3
14	PART8	PART8*	2.6	2.6
15	PART9	PART9*	7.8	7.8

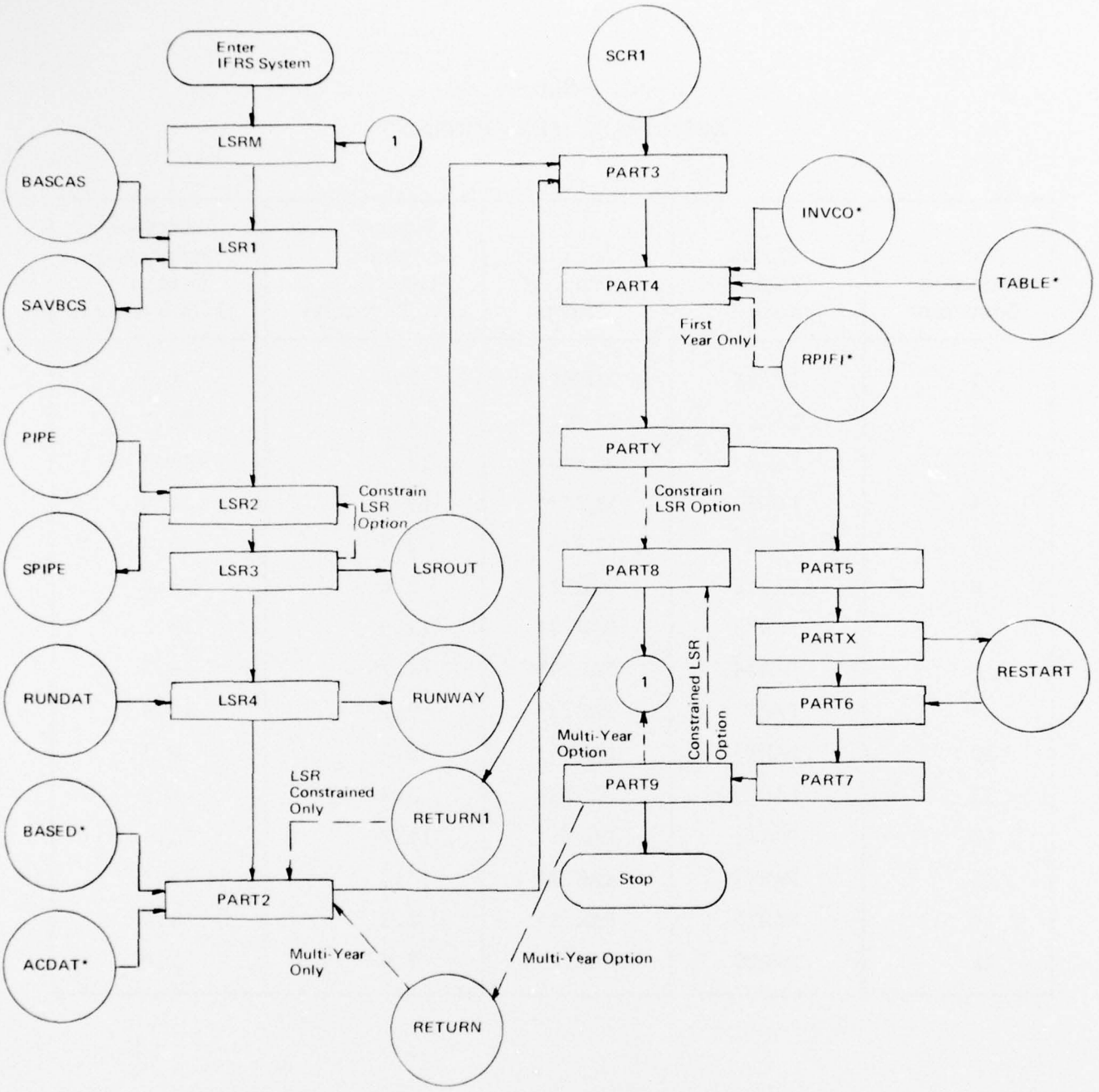


FIGURE 1. OVERVIEW OF IFRS COMPUTER PROGRAM

TABLE 2

COMPUTER PROGRAMS

Name	Type	Description
LSRM	Program	Enters level of complexity and annual data used in LSR Generator
LSR1	Program	Inputs and modifies LSR training phase data file
BASCAS	Data File	Training phase data used by LSR Generator
SAVBCS	Data File	Modified training phase data
LSR2	Program	Computes student statistics for all training pipeline
PIPE	Data File	Pipeline data for each training pipeline
SPIPE	Data File	Modified pipeline data
LSR3	Program	Computes LSR summary statements
LSROUT	Data File	LSR summary statements
LSR4	Program	Develops runway requirements
RUNDAT	Data File	Runway data for each training phase
RUNWAY	Data File	Runway requirements
PART2	Program	Reads Base Data File, Aircraft Data File, initializes on various returns from LSR Generator
PART3	Program	Reads LSR data by phase, records phase allocations, and aggregates LSR data by base
BASED*	Data File	Base dependent parameters
ACDAT*	Data File	Aircraft data
RETURN	Data File	Updated inventory for multi-year runs
RETURN1	Data File	Phase allocation stored for regeneration upon return from LSR

TABLE 2 (Cont)

Name	Type	Description
SCRI	Scratch File	Used to pick off first character of fuel type names
PART4	Program	Reads data files, processes OLF and runway requirements, air saturation
INVCO*	Data File	Cost data for facilities
TABLE*	Data File	Various tables for computing facility requirements
RPIFI*	Data File	Inventory of facilities for all bases
PARTY	Program	Computes runway deficits and costs
PART5	Program	Computes facility requirements
PARTX	Program	Places common variables in restart file
RESTART	Data File	Provides restart procedure immediately preceding excess/deficiency
PART6	Program	Excess/deficiency computations
PART7	Program	Computes facility and aircraft investment costs
PART8	Program	Records present phase allocation prior to exit to LSR Generator
PART9	Program	O&M cost computations, total systems cost

II. PROGRAM LSRM [1/]

PROGRAM DESCRIPTION

2.1 The purpose of PROGRAM LSRM is to initialize program variables for the LSR Generator. PROGRAM LSRM is the first computer program to be run in the automated IFRS system. It may be entered from three sources: (a) directly by the user from the time-sharing terminal; (b) from PROGRAM PART8 when constraints are imposed on a previous output of the LSR Generator; or (c) from PROGRAM PART9 when LSR's output from another year is desired. Upon entry, the eleventh position in common, IS(7) (named ISWTCH(10) in PROGRAMS PART8 and PART9), is tested for zero.

2.2 When IS(7) = 0, entry into PROGRAM LSRM comes from the user, making it the first entry into the computer program. The level of LSR complexity, LEVLSR; annual fly days, AFD; and annual training weeks, WPY; are entered by the user. These values are checked for validity and stored in the second,third, and fourth common locations—ISW, SW(1),and SW(2), respectively.

2.3 Next, or when IS(7) > 0, the values of LEVLSR, AFD, and WPY are set equal to their respective saved common values. When IS(7) = 2, the LSR output is to be constrained, and the level of complexity, LEVLSR, is negated for subsequent reference. Control is transferred to PROGRAM LSR1. Note that the first 11 common locations are used for inter-program linkages. All other stored data may be renamed and used for other storage locations by all IFRS computer programs.

2.4 A flow chart of PROGRAM LSRM is shown in Figure 2. Table 3 contains the variable dictionary of PROGRAM LSRM; the program and subroutine dictionary is provided in Table 4. The program listing is shown in Table 5.

[1/] Programs LSRM, LSR1, LSR2, and LSR3 constitute the LSR Generator discussed in Volumes I and II.

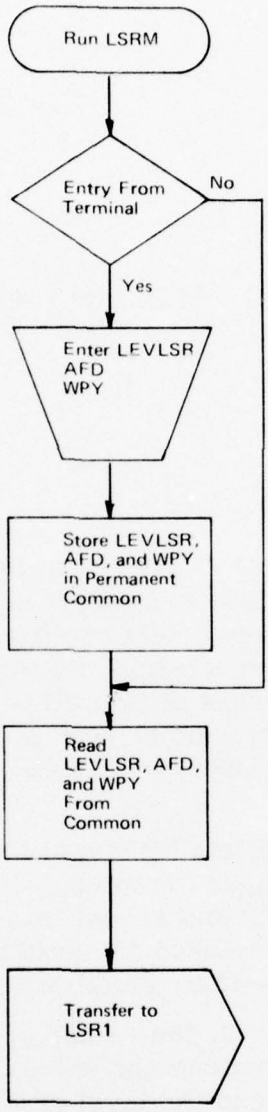


FIGURE 2. PROGRAM LSRM FLOW CHART

TABLE 3

PROGRAM LSRM VARIABLE DICTIONARY

Location	Variable Name	Dimension	Type [1]	Description
Common	IY	1	I	Year—not used
Common	ISW	1	I	Permanent storage for the level of complexity for the LSR Generator
Common	SW	2	F	SW(1): permanent storage for annual fly days SW(2): permanent storage for training weeks per year
Common	IS [2]	7	I	IS(1) to ISW(6): permanent storage not used IS(7) = 0: first entry into PROGRAM LSRM IS(7) = 1: generate an LSR output for the next year IS(7) = 2: constrain LSR output for current year
Common	NAME	25,3	A	Name of training phase I [3] (3 words or 12 characters permitted)
Common	NPLA	25,3	A	Name of aircraft types for phase I, J = 1, 3 [4] denotes up to 3 aircraft types
Common	NFUEL	25,3	A	Fuel type for phase I, aircraft type J
Common	NACD	25,3	A	Academic instruction types for phase I
Common	ATP	25	F	Average portion of phase I a student attrite completes
Common	WK	25	F	Instructor tour of duty length for phase I
Common	NAC	25	I	Number of aircraft types for phase I (must not exceed 3)

[1] Variable type I refers to integer, F to floating point; A to alphameric variables.

[2] Note the above 11 common locations are used for permanent storage. They are not overlayed or used in a different capacity by any of the 15 IFRS programs.

[3] I refers to row dimension.

[4] J refers to column dimension.

TABLE 3 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	NAD	25	I	Number of academic instruction types for phase I (≤ 3)
Common	WX	25,3	F	Percent flyable weather for aircraft type J in phase I
Common	GAS	25,3	F	Fuel consumption rate for aircraft type J in phase I
Common	AV	25,3	F	Daily aircraft utilization for aircraft type J in phase I
Common	FU	25,3	F	Daily flight instructor utilization for flight instructor type J in phase I
Common	SFH	25,3	F	Student flight hours to complete a successful student in flight instruction type J in phase I
Common	FIH	25,3	F	Flight instructor hours required for a successful student to complete flight training type J in Phase I
Common	FTR	25,3	F	Flight instructor training period for instructor type J in phase I
Common	FSO	25,3	F	Landing support officer to student ratio for flight instruction type J in phase I
Common	AMO	25,3	F	Enlisted maintenance personnel per aircraft type J in phase I
Common	ASH	25,3	F	Student academic hours for academic instruction type J in phase I
Common	AIH	25,3	F	Academic instructor hours for academic instruction type J in phase I
Common	AITR	25,3	F	Academic instructor training period for academic instruction type J in phase I
Common	ICOMMA	1	A	Comma ","

TABLE 3 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	IBLANK	1	A	Space " "
Common	NO	1	A	Letter N "N"
Common	NYES	1	A	Letter Y "Y"
Common	NY	1	I	Switch for Yes-No input NY = -1 previous response No, "N" = 1 previous response Yes, "Y"
Common	NPH	1	I	Number of training phases (≤ 25)
Common	IER	1	I	Error type switch
Common	LEVLSR	1	I	Level of complexity for LSR Generator
Common	IPH	1	I	Phase number of particular training phase
Common	WPY	1	F	Training weeks per year
Common	AFD	1	F	Annual fly days
Common	KILL	1	I	Number of total training phases deleted in current run from data base
Common	IID	1	I	Temporary storage for subroutine transfer
Common	FID	1	F	Temporary storage for subroutine transfer
Common	KILLS	25	I	Phase numbers of deleted phases
Common	SI	25	F	Student input for all pipelines
Common	TSOUT	25	F	Student output for all pipelines
Common	SO	25	F	Student output for particular pipeline
Common	IBC	1	I	Base case switch = 0: base case data have not been modified = 1: base case data have been modified

TABLE 3 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	IL	1	I	Line number on output file and temporary storage
Common	IP	1	I	Element number for modified phase data
Common	N	1	I	Number of phases to be deleted or listed
Common	ITEM	1	I	Item number of phase data to be tested for validity
Common	IDEL	51	I	Numbers of phases to be deleted or listed. Note even numbered fields contain a ","
Common	BMAX	15	F	Maximum expected values for training phase data

TABLE 4

PROGRAM LSRM PROGRAM AND SUBROUTINE DICTIONARY

LSRM	Main program in automated IFRS system. Initializes program variables for the LSR Generator.

TABLE 5

PROGRAM LSRM LISTING

```
100       C0MM0N IY,ISW,SW(2),IS(7)
120       C0MM0N NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
140      &WK(25),T0D(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
160      &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FS0(25,3),AM0(25,3),
180      &ASH(25,3),AIH(25,3),AITR(25,3)
200       C0MM0N IC0MMA,IBLANK,N0,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
220      &AFD,KILL,IID,FID,KILLS(25),SI(25),TS0UT(25),S0(25)
240       C0MM0N IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
260       KILL=0
280       IBC=0
300       IF(IS(7).NE.0)G0 T0 200
320     5 PRINT 700
340    10 INPUT 701,LEVLSR
360       IF(LEVLSR)30,30,20
380    20 IF(LEVLSR-4)40,40,30
400    30 PRINT 702
420       G0 T0 10
440    40 PRINT 703
460    50 INPUT,WPY,AFD
480       IF(WPY)90,90,60
500    60 IF(WPY-52.)70,70,90
520    70 IF(AFD)90,90,80
540    80 IF(AFD-365.)100,100,90
560    90 PRINT 702
580       G0 T0 50
600   100 ISW=LEVLSR
620       SW(1)=AFD
640       SW(2)=WPY
660   200 LEVLSR=ISW
680       IF(LEVLSR.EQ.0)G0 T0 5
700       AFD=SW(1)
720       WPY=SW(2)
740       IF(IS(7).EQ.2)LEVLSR=-LEVLSR
760       CHAIN "XLSR1*"
780   700 F0RMAT(26H ENTER LEVEL 0F C0MPLEXITY/34H 1 N0 ADJUSTMENT
800      &S 0R M0DIFICATI0NS/23H 2 C0NSTRAIN LSR 0UTPUT/20H 3 M0DIFY
820      & PHASE DATA/26H 4 C0MBINE 0PTI0NS 2 AND 3/)
840   701 F0RMAT(I1)
860   703 F0RMAT(" ENTER TRAINING WEEKS PER YEAR"/" AND ANNUAL FLY-
880      &DAYS (XX.,XXX.)")
900   702 F0RMAT(23H INVALID REPLY - REPEAT)
920       END
```

III. PROGRAM LSR1

PROGRAM DESCRIPTION

3.1 The purpose of PROGRAM LSR1 is to input into memory the data associated with each training phase and to update this data base when applicable. Upon entry, a test is made on Switch IBC to determine whether the data bse has been modified in a previous year's run. Note that the automated IFRS system can be run sequentially to determine total systems cost for several years. If Switch IBC = 0, the normal base case data file, BASCAS, is accessed. However, should IBC = 1, the user is asked whether the base case data can be restored (BASCAS accessed) or the previously modified data base, SAVBCS, accessed.

3.2 The accessed file is then opened and rewound. The phase data for each training phase are sequentially read from the data file. After the phase data have been read for an individual training phase, a transfer is made to Subroutine CHECKP to check the data for validity, e.g., the percent of flyable weather may not exceed 100 percent, etc.

3.3 The user is then asked whether he desires a list of the training phases. With a Yes response, "Y," transfer is made to Subroutine PHASES to develop this data list. The accessed phase data file is then closed. Next a test is made to determine whether the user desires to modify the phase data (LEVLSR = 3 or 4). If this option is specified, Subroutine MODIFY is entered to revise the phase data. Control passes to PROGRAM LSR2.

SUBROUTINE MODIFY

3.4 The purpose of Subroutine MODIFY is to update the training phase data base. Upon entry, the user is asked whether he wishes to delete a training phase

or add a new phase. With a Yes response, "Y," the user is then asked whether any training phases are to be deleted. With another Yes response, "Y," control passes to Subroutine DELETE to delete training phases.

3.5 The user is asked, when applicable, if new training phases are to be added. When a Yes reply is given, the number of training phases, NPH, is incremented by 1 (not to exceed the 25 maximum) and control is passed to Subroutine NEWPHA. This process is continued until no further phases are to be added. After all new phases are added, Subroutine PHASES is called to list the phases which currently exist.

3.6 The user is then asked whether he wishes to list the phase data or make modifications. If the response is "Yes," he is asked whether any phase data lists are desired. With a Yes reply, Subroutine EDIT1 is entered to input the numbers of the phases which are to be listed. Subroutine LIST is then called, once for each item of each phase number entered, to print the phase data.

3.7 Next the user is asked whether any training phase data are to be modified. If training phase data are to be updated, the user specifies the training phase, the data field, and the element to be revised. The new data element is entered in Subroutine UPDATE. Note that if the number of aircraft or academic instruction types are increased, all data associated with them must be entered. For example, if the number of aircraft is increased, the aircraft names, fuel names, weather factor, daily utilization rates, etc., must be entered. After a data point has been entered, the revised value is automatically listed.

3.8 The subroutine cycles, asking the user whether he wishes a data list or wants to modify the phase data, until a No, "N," response has been received. With a No reply, all training phase data are tested for validity through Subroutine CHECKP. The user is then asked whether this new data base should be saved. When revised data are to be temporarily stored in data file SAVBCS, the modified phase data file includes this modified data base. Control returns to PROGRAM LSR1.

SUBROUTINE CHECKP

3.9 The purpose of Subroutine CHECKP is to develop the appropriate routine linkage to test all phase data for validity. The procedure used is to store sequentially each phase data point in common locations IID or FID, depending on whether the individual data element is an integer (IID) or floating point (FID) number. An item number, ITEM, is then updated to reflect the particular data element which is to be tested. For example, ITEM = 1 implies the data element is the attrition point; ITEM = 2 refers to weeks duration of the training phase, etc. Subroutine DTEST is then called to validate the particular data point. Subroutine CHECKP tests the 15 floating point and 2 integer data values for each training phase.

SUBROUTINE NOYES

3.10 Subroutine NOYES reads a Yes, "Y," or No, "N," response from the time-sharing terminal. Switch NY, which is in common, is set to -1 for a No response and to 1 for a Yes response. When a Yes or No reply is required in PROGRAM LSR1 or its subroutines, Subroutine NOYES is called.

SUBROUTINE DELETE

3.11 Subroutine DELETE deletes training phases from the training phase data base. Upon entry, Subroutine EDIT4 is called which inputs the numbers of the training phases which are to be deleted. The phases are deleted in descending order and thus the phase numbers of the training phases to be deleted are scanned to determine the maximum number, i.e., if Phases 4 and 8 are to be deleted, Phase 8 is deleted first. A list of these phase numbers and names is printed for reference. The deleted phase number is stored in variable KILLS for reference in PROGRAM LSR2. To provide program consistency in all routines, the remaining phases are renumbered to keep the phase numbers sequential starting at number 1. For example, if 15 training phases exist in the pipeline and Phase 8 is deleted, old Phase 9 would become Phase 8, Phase 10 becomes 9, etc.

3.12 Subroutine DELETE continues the foregoing process, i.e., select the next highest phase number until the specified phase numbers have been deleted. Control is then returned to Subroutine MODIFY, the calling routine.

SUBROUTINE NEWPHA

3.13 Subroutine NEWPHA develops the appropriate linkage to add a new training phase into the training phase data base. Upon entry, a transfer is made to Subroutine UPDATE and the first six training phase data points are entered, i.e., phase name, through number of academic instructions. If flight instruction is specified, NAC > 0. Subroutine UPDATE is called to enter the data associated with aircraft flight instruction (training phase items 7 to 17). When academic instruction is specified, NAD > 0, training phase items 18 to 21 are similarly entered.

3.14 When all data associated with a training phase have been entered, Subroutine LIST is entered to print these new data. Control then returns to Subroutine MODIFY.

SUBROUTINE EDIT1

3.15 Subroutine EDIT1 reads from the time-sharing terminal the phase numbers of the training phases which are to be deleted or listed. Upon entry, the user provides the appropriate phase numbers depending on the particular option being exercised. A check is then made to determine the validity of the phase numbers entered. The number of entries, N, is developed and control is returned to the calling routine (Subroutine MODIFY).

SUBROUTINE LIST

3.16 Subroutine LIST prints the phase data associated with a particular training phase, IPH. Upon entry, an appropriate branch is made to the print statement which will list the phase data associated with a particular item, IL. For example, if IL = 5, the number of different aircraft types for the particular training phase is listed. Note that each training phase may contain up to 21 different types of data. To obtain a complete data list, i.e., all data for a particular training phase, Subroutine LIST is called 21 times with IL being serially incremented from 1.

SUBROUTINE UPDATE

3.17 Subroutine UPDATE inputs from the user a particular training phase data value. Upon entry, a test is made to determine which data point is to be entered. The program branches to the appropriate input statement so that revised values may be user entered. For non-alphameric data fields, Subroutine DTEST is called to validate the user supplied value. Control then returns to the calling routine.

SUBROUTINE DTEST

3.18 Subroutine DTEST checks a particular phase data point for validity. Upon entry, a check is made to determine whether the data point to be tested is a floating point number. If it is a floating point number, the value is checked against the appropriate maximum expected value, BMAX. If the value exceeds BMAX, a message is printed for the user identifying the data element. The user either accepts the data value or enters a new value.

3.19 A similar procedure is used for integer data points. However, since the only integer variables in the phase data base are the number of flight and academic instruction types, the maximum value is 3. Note that this constraint arises because aircraft and academic instruction types have a program dimension of 3. Control then returns to the calling routine.

SUBROUTINE PHASES

3.20 Subroutine PHASES lists the name and number of each training phase contained in the data base.

SUBROUTINE ERROR

3.21 Subroutine ERROR prints diagnostic messages. In accordance with an error type Switch, IER, a diagnostic message is displayed.

3.22 A flow chart of PROGRAM LSR1 is shown in Figure 3. Table 6 contains the variable dictionary of PROGRAM LSR1; the program and subroutine dictionary is provided in Table 7. The program listing is shown in Table 8.

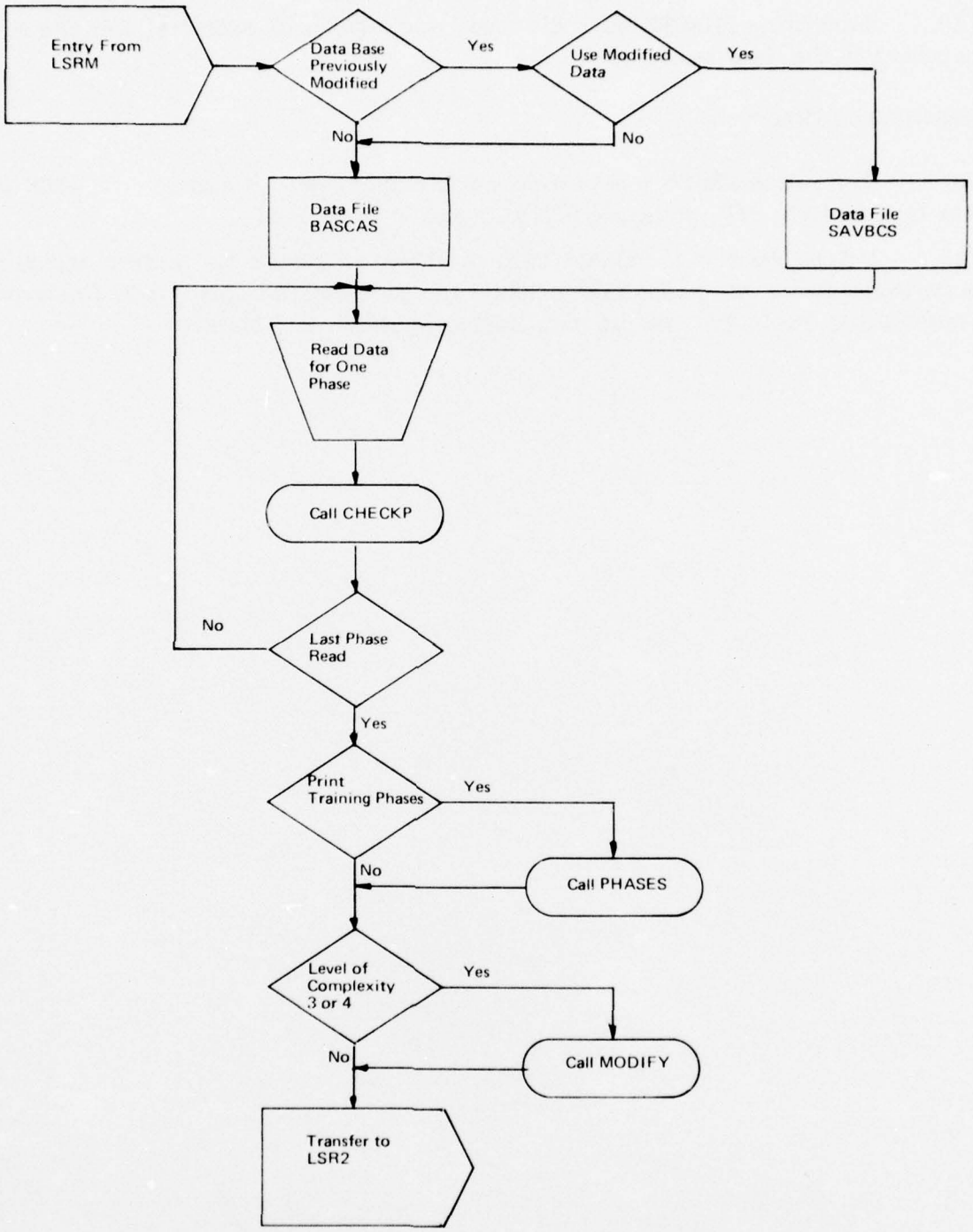


FIGURE 3. PROGRAM LSR1 FLOW CHART

a. Subroutine MODIFY

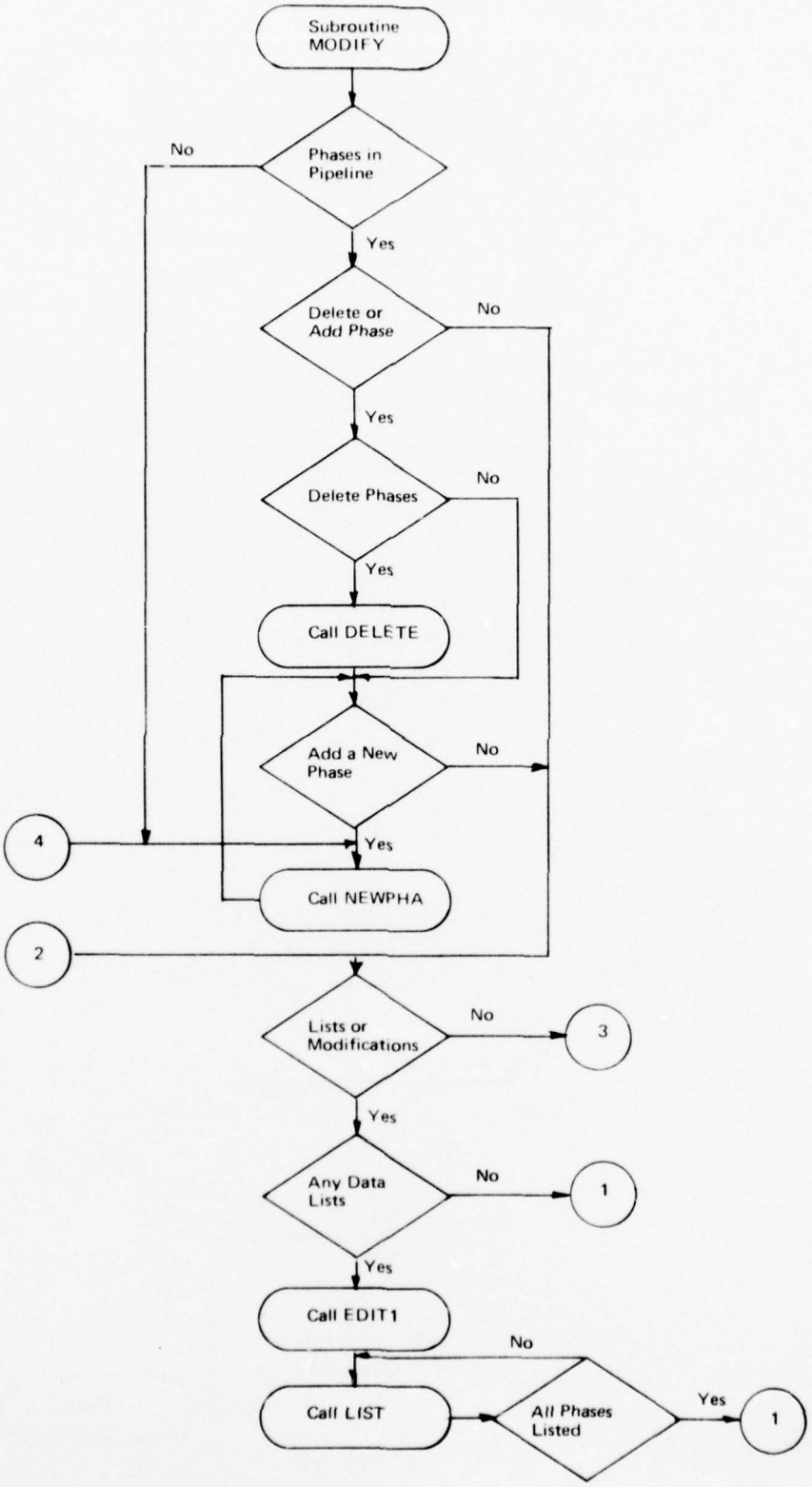


FIGURE 3 (Cont)

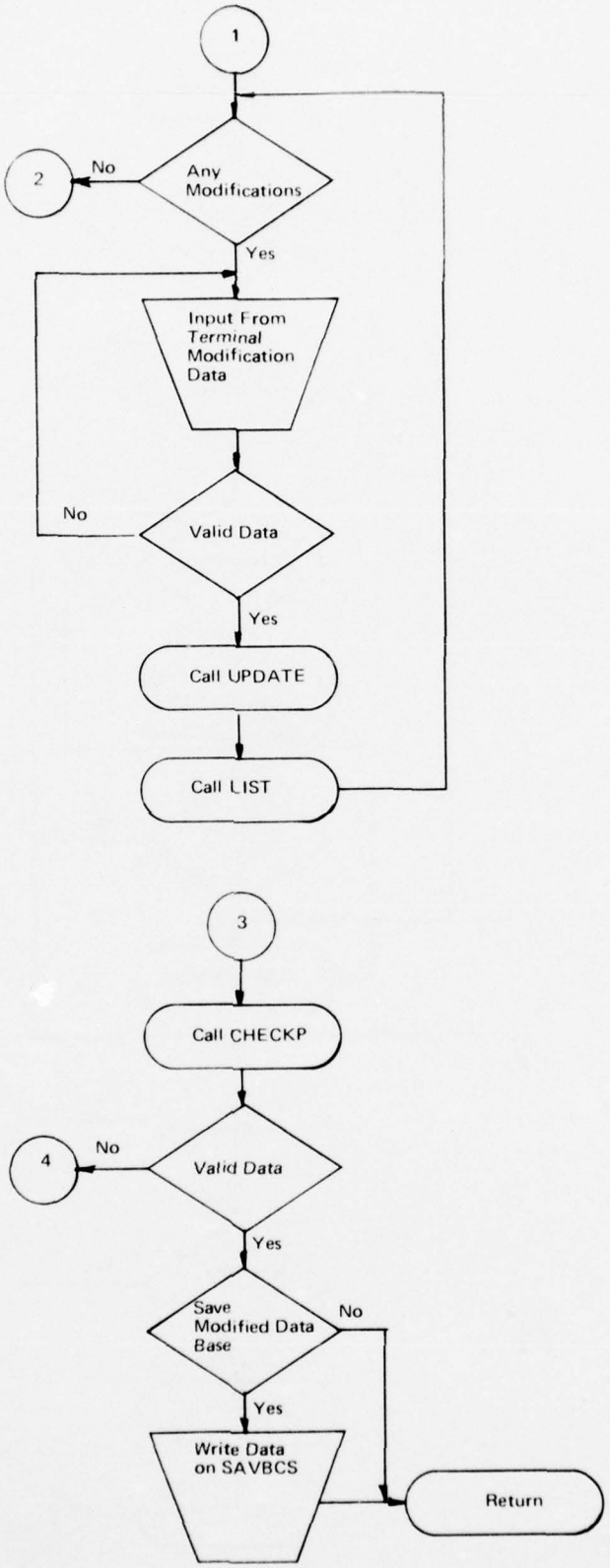


FIGURE 3 (Cont)

b. Subroutine CHECKUP

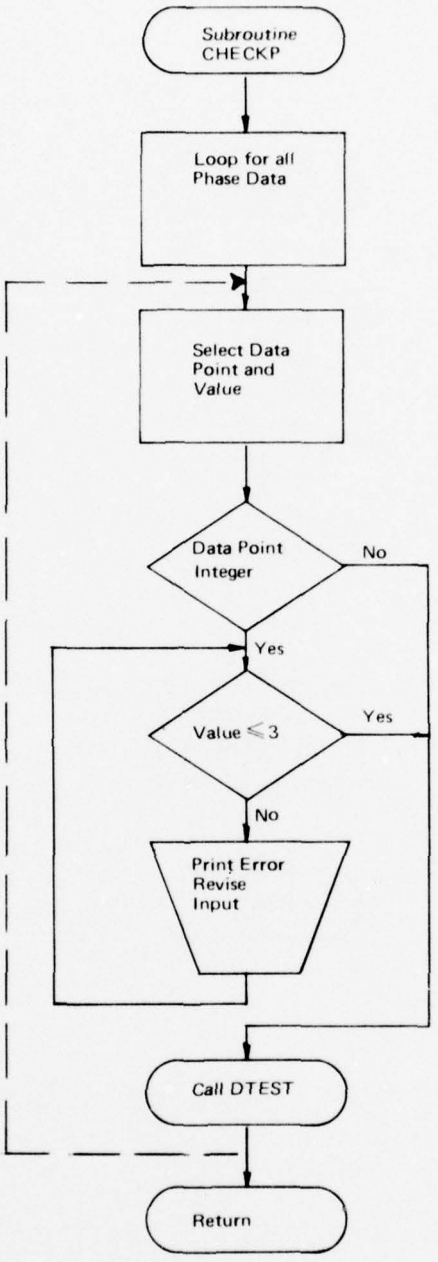


FIGURE 3 (Cont)

c. Subroutine NOYES

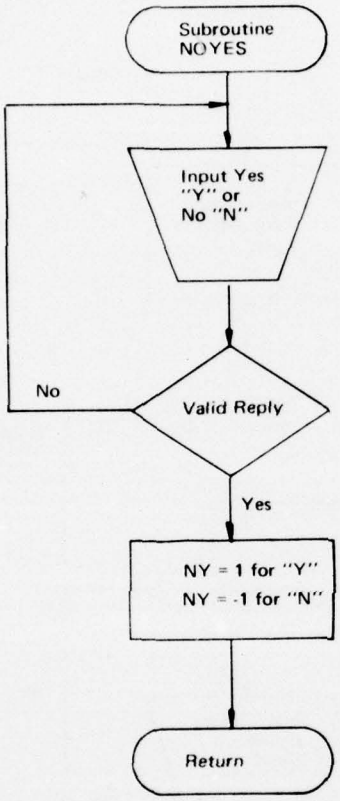


FIGURE 3 (Cont)

d. Subroutine DELETE

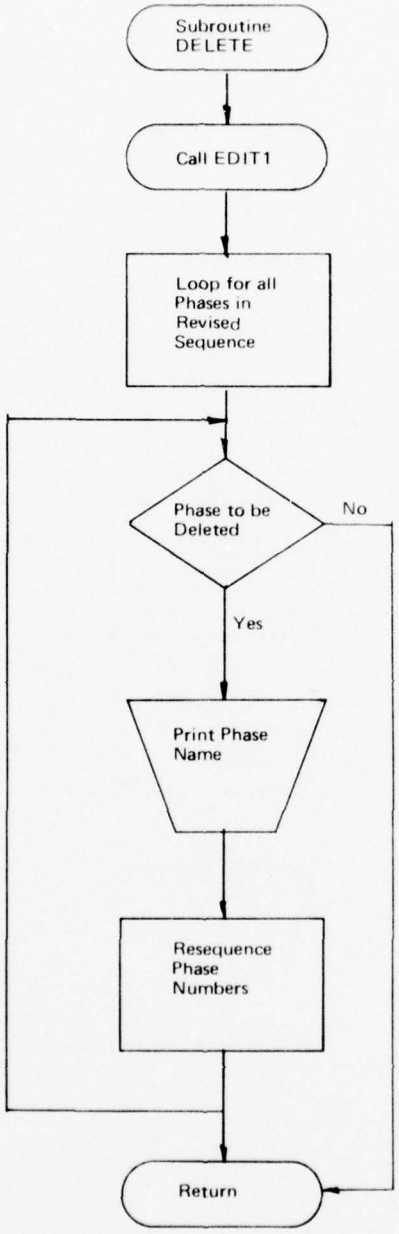


FIGURE 3 (Cont)

e. Subroutine NEWPHA

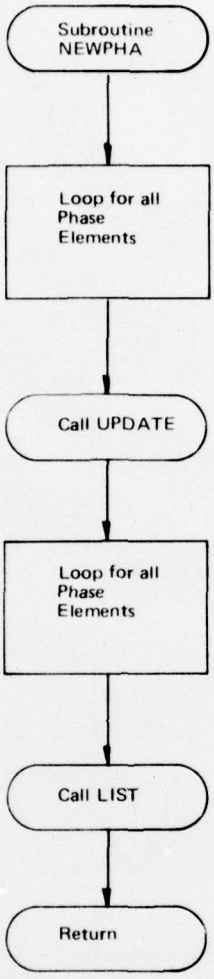


FIGURE 3 (Cont)

f. Subroutine EDIT1

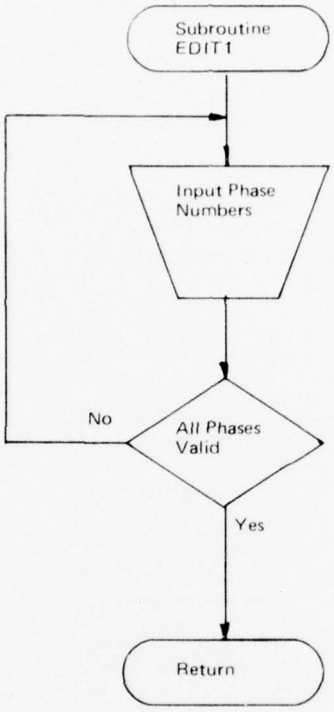
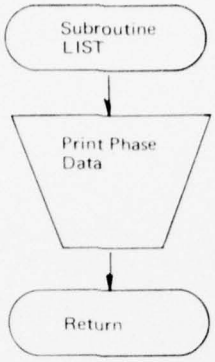
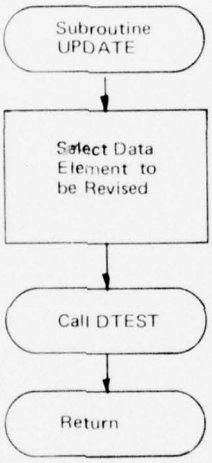


FIGURE 3 (Cont)

g. Subroutine LIST



h. Subroutine UPDATE



i. Subroutine DTEST

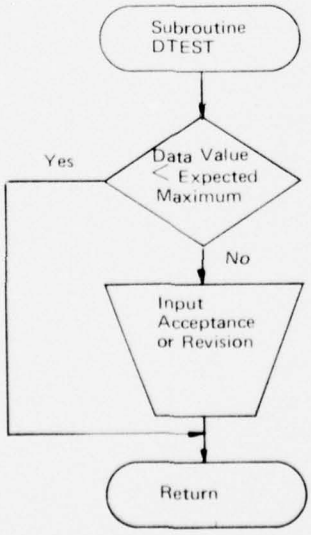
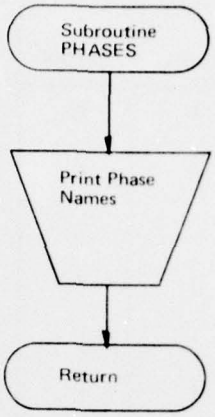


FIGURE 3 (Cont)

j. Subroutine PHASES



k. Subroutine ERROR

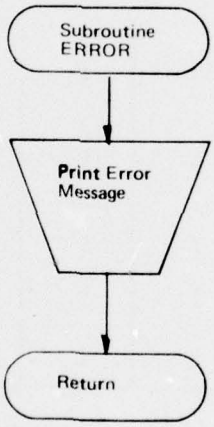


FIGURE 3 (Cont)

TABLE 6

PROGRAM LSR1 VARIABLE DICTIONARY*

Location	Variable Name	Dimension	Type	Description
LSR1	INP	1	A	File name
MODIFY	IS	1	I	New phase switch IS = 0: no new phases added IS = 1: new phases have been added
MODIFY	ILB	1	I	Lower print loop bound
MODIFY	IUB	1	I	Upper print loop bound
MODIFY	K	1	I	Number of aircraft before aircraft data field was modified
MODIFY	N	1	I	Revised number of aircraft
CHECKP	IAD	25,3,4	A	Equivalent to common variable NAME
CHECKP	DF1	25,3	F	Equivalent to common variable ATP
CHECKP	DF2	25,3,12	F	Equivalent to common variable GAS
DELETE	IDF1	25,2	I	Equivalent to common variable NAC
LIST	NACC	1	I	Number of aircraft types
LIST	NADD	1	I	Number of academic instruction types

* For a dictionary of common variables, see Table 3.

TABLE 7

PROGRAM LSR1 PROGRAM AND SUBROUTINE DICTIONARY

LSR1	Inputs all data associated with each phase of pilot training.
MODIFY	Modifies training phase data by deleting phases, adding new phases, listing existing phase data, and modifying phase data. The modified data base is stored in file SAVBCS.
CHECKP	Scans each training phase for valid data.
NOYES	Reads a Yes, "Y," or No, "N," response from the time-sharing terminal.
DELETE	Deletes training phases from the data base.
NEWPHA	Adds a new training phase into the data base.
EDIT1	Reads from the time-sharing terminal the numbers of phases to be deleted or data listed.
LIST	Lists all data associated with a phase of pilot training.
UPDATE	Modifies each data field with respect to a particular phase of training.
DTEST	Tests a particular data point for validity. Modifications to invalid data points are made when requested.
PHASES	Lists the number and name of each training phase in the current data base.
ERROR	Prints error messages when invalid terminal responses arise.

TABLE 8

PROGRAM LSR1 LISTING

```
LSR1          14:18       01/18/70

101         COMMON IYEAR,ISWTCH(10)
121         COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
141        &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
161        &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
181        &ASH(25,3),AIH(25,3),AITR(25,3)
201         COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
221        &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
241         COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
261         FILENAME INP
281         IF(IBC)5,5,100
301       5 INP="BASCAS"
321      10 OPENFILE INP
341         REWIND INP
361         READ(INP,700)NO,NYES,ICOMMA,IBLANK
381         READ(INP,701)IL,BMAX
401         READ(INP,701)IL,NPH
421         IF(NPH)90,90,20
441      20 IF(NPH-25)30,30,90
461      30 DO 40 I=1,NPH
481         READ(INP,703)(NAME(I,J),J=1,3),(NPLA(I,J),J=1,3),
501        &(NFUEL(I,J),J=1,3),(NACD(I,J),J=1,3)
521         READ(INP,701)IL,NAC(I),NAD(I)
541         READ(INP,701)IL,ATP(I),WK(I),TOD(I)
561         READ(INP,701)IL,(WX(I,J),J=1,3)
581         READ(INP,701)IL,(GAS(I,J),J=1,3)
601         READ(INP,701)IL,(AU(I,J),J=1,3)
621         READ(INP,701)IL,(FU(I,J),J=1,3)
641         READ(INP,701)IL,(SFH(I,J),J=1,3)
661         READ(INP,701)IL,(FIH(I,J),J=1,3)
681         READ(INP,701)IL,(FTR(I,J),J=1,3)
701         READ(INP,701)IL,(FSO(I,J),J=1,3)
721         READ(INP,701)IL,(AMO(I,J),J=1,3)
741         READ(INP,701)IL,(ASH(I,J),J=1,3)
761         READ(INP,701)IL,(AIH(I,J),J=1,3)
781         READ(INP,701)IL,(AITR(I,J),J=1,3)
801         IPH=I
821         CALL CHECKP
841      40 CONTINUE
861         IF(NPH)90,90,50
881      50 PRINT 705
901         CALL NOYES
921         IF(NY)80,80,60
941      60 CALL PHASES
```

TABLE 8 (Cont)

```
961       KILL=0
981    80 CLOSEFILE INP
1001       IF(LEVLSR-2)87,87,83
1021    83 CALL MODIFY
1041    87 CHAIN "XLSR2*"
1061    90 NPH=0
1081       IER=3
1101       CALL ERROR
1121       LEVLSR=4
1141       PRINT 706
1161       GO TO 80
1181   100 PRINT 707
1201       CALL NOYES
1221       IF(NY)110,110,105
1241   105 IBC=0
1261       GO TO 5
1281   110 INP="SAVBCS"
1301       GO TO 10
1321   700 FORMAT(5X,3A1,A4)
1341   701 FORMAT(V)
1361   703 FORMAT(5X,12A4)
1381   705 FORMAT(" PRINT LIST OF TRAINING PHASES (Y,N)")
1401   706 FORMAT(31H LSR COMPLEXITY OPTION SET TO 4)
1421   707 FORMAT(24H RESTORE BASE CASE (Y,N))
1441       END
```

TABLE 8 (Cont)

a. Subroutine MODIFY

```
1461        SUBROUTINE MODIFY
1481        COMMON IYEAR,ISWTCH(10)
1501       . COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
1521       &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
1541       &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
1561       &ASH(25,3),AIH(25,3),AITR(25,3)
1581        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
1601       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
1621        COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
1641        FILENAME OUT
1661        OUT="SAVBCS"
1681        IS=0
1701        IF(NPH)90,90,10
1721     10 PRINT 700
1741        CALL NOYES
1761        IF(NY)120,120,20
1781     20 PRINT 701
1801        CALL NOYES
1821        IF(NY)40,40,30
1841     30 CALL DELETE
1861     40 PRINT 702
1881        CALL NOYES
1901        IF(NY)50,50,70
1921     50 IF(NPH)60,60,100
1941     60 IER=3
1961        NPH=0
1981        CALL ERROR
2001     70 IF(NPH-25)90,80,80
2021     80 IER=4
2041        CALL ERROR
2061        GO TO 20
2081     90 NPH=NPH+1
2101        IPH=NPH
2121        IS=1
2141        CALL NEWPHA
2161        GO TO 40
2181    100 IF(IS)120,120,110
2201    110 CALL PHASES
2221    120 PRINT 715
2241        CALL NOYES
2261        IF(NY)500,500,125
2281    125 PRINT 703
2301        CALL NOYES
```

TABLE 8 (Cont)

a. Subroutine MODIFY (Cont)

```
2321        IF(NY)180,180,130
2341   130  CALL EDIT1
2361        IF(N)150,150,160
2381   150  IER=2
2401        CALL ERROR
2421        GO TO 180
2441   160  DO 170 I=1,N,2
2461        IPH=IDEL(I)
2481        DO 170 J=1,22
2501        IL=J-1
2521        CALL LIST
2541   170  CONTINUE
2561   180  PRINT 704
2581        CALL NOYES
2601        IF(NY)120,120,190
2621   190  PRINT 705
2641   200  INPUT 706,IPH,IC1,IL,IC2,IP
2661        IF(IPH)210,120,220
2681   210  IER=6
2701   215  CALL ERROR
2721        GO TO 200
2741   220  IF(IPH-NPH)230,230,210
2761   230  IF(IC1-ICOMMA)240,250,240
2781   240  IER=1
2801        GO TO 215
2821   250  IF(IL)210,210,260
2841   260  IF(IL-5)270,290,330
2861   270  CALL UPDATE
2881        CALL LIST
2901   280  PRINT 707
2921        GO TO 200
2941   290  K=NAC(IPH)
2961        ILB=7
2981        IUB=17
3001        CALL UPDATE
3021        CALL LIST
3041        N=NAC(IPH)
3061   300  IF(K-N)310,280,280
3081   310  K=K+1
3101        DO 325 I=ILB,IUB
3121        IL=I
3141        DO 320 J=K,N
3161        IP=J
```

TABLE 8 (Cont)

a. Subroutine MODIFY (Cont)

```
3181   320 CALL UPDATE
3201   325 CALL LIST
3221       GO TO 280
3241   330 IF(IL-6)340,340,350
3261   340 K=NAD(IPH)
3281       ILB=18
3301       IUB=21
3321       CALL UPDATE
3341       CALL LIST
3361       N=NAD(IPH)
3381       GO TO 300
3401   350 N=NAC(IPH)
3421       IF(IL-17)360,360,390
3441   360 IF(IP)210,210,380
3461   380 IF(IP-N)270,270,210
3481   390 N=NAD(IPH)
3501       IF(IL-21)360,360,210
3521   500 DO 510 I=1,NPH
3541       IPH=I
3561   510 CALL CHECKP
3581       IF(NPH)90,90,530
3601   530 PRINT 708
3621       CALL NOYES
3641       IF(NY)560,560,540
3661   540 IBC=1
3681       OPENFILE OUT
3701       REWIND OUT
3721       WRITE(OUT,709)NO,NYES,ICOMMA,IBLANK
3741       WRITE(OUT,710)BMAX
3761       WRITE(OUT,711)NPH
3781       IC=1025
3801       DO 550 I=1,NPH
3821       IC=IC+5
3841       WRITE(OUT,712)IC,(NAME(I,J),J=1,3),(NPLA(I,J),J=1,3),
3861      &(NFUEL(I,J),J=1,3),(NACD(I,J),J=1,3)
3881       IC=IC+5
3901       WRITE(OUT,713)IC,NAC(I),NAD(I)
3921       IC=IC+5
3941       WRITE(OUT,714)IC,ATP(I),WK(I),TOD(I)
3961       IC=IC+5
3981       WRITE(OUT,714)IC,(WX(I,J),J=1,3)
4001       IC=IC+5
4021       WRITE(OUT,714)IC,(GAS(I,J),J=1,3)
```

TABLE 8

a. Subroutine MODIFY (Cont)

```
      IC=IC+5
      WRITE(OUT,714)IC,(AU(I,J),J=1,3)
      IC=IC+5
      WRITE(OUT,714)IC,(FU(I,J),J=1,3)
      IC=IC+5
      WRITE(OUT,714)IC,(SFH(I,J),J=1,3)
      IC=IC+5
      WRITE(OUT,714)IC,(FIH(I,J),J=1,3)
      IC=IC+5
      WRITE(OUT,714)IC,(FTR(I,J),J=1,3)
      IC=IC+5
      WRITE(OUT,714)IC,(FSO(I,J),J=1,3)
      IC=IC+5
      WRITE(OUT,714)IC,(AMO(I,J),J=1,3)
      IC=IC+5
      WRITE(OUT,714)IC,(ASH(I,J),J=1,3)
      IC=IC+5
      WRITE(OUT,714)IC,(AIH(I,J),J=1,3)
      IC=IC+5
      WRITE(OUT,714)IC,(AITR(I,J),J=1,3)
  550 CONTINUE
      CLOSEFILE OUT
  560 RETURN
  700 FORMAT(//33H ANY DELETIONS OR ADDITIONS (Y,N))
     &N))
  701 FORMAT(20H ANY DELETIONS (Y,N))
  702 FORMAT(22H ADD A NEW PHASE (Y,N))
  703 FORMAT(21H ANY DATA LISTS (Y,N))
  704 FORMAT(24H ANY MODIFICATIONS (Y,N))
  705 FORMAT(41H ENTER PHASE, FIELD AND ELEMENT (XX,XX,X)/44H PHASE
     & = 00 IMPLIES NO FURTHER MODIFICATIONS/" NOTE TWO DIGIT
     &FIELDS MUST CONTAIN TWO DIGITS")
  706 FORMAT(2(I2,A1),I1)
  707 FORMAT(5H NEXT)
  708 FORMAT(30H SAVE MODIFIED DATA BASE (Y,N))
  709 FORMAT(5H1000 ,3A1,A4)
  710 FORMAT(5H1005 ,4E13.6/5H1010 ,4E13.6/5H1015 ,4E13.6/
     &5H1020 ,4E13.6)
  711 FORMAT(5H1025 ,I3)
  712 FORMAT(I4,1X,12A4)
  713 FORMAT(I4,1X,2I3)
  714 FORMAT(I4,1X,3E13.6)
  715 FORMAT(" ANY LISTS OR MODIFICATIONS (Y,N)")
      END
```

TABLE 8 (Cont)

b. Subroutine CHECKUP

```
4921        SUBROUTINE CHECKP
4941        COMMON IYEAR,ISWTCH(10)
4961        COMMON IAD(25,3,4),DF1(25,3),NAC(25),NAD(25),DF2(25,3,12)
4981        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
5001       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
5021        COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
5041        DO 20 I=1,3
5061        IL=I-1
5081        ITEM=I
5101        FID=DF1(IPH,I)
5121        CALL DTEST
5141     20 DF1(IPH,I)=FID
5161        N=NAC(IPH)
5181        IF(N)40,80,30
5201     30 IF(N-3)50,50,40
5221     40 IL=0
5241        CALL LIST
5261        IL=5
5281        CALL LIST
5301        NAC(IPH)=0
5321        IER=5
5341        CALL ERROR
5361        GO TO 80
5381     50 DO 70 I=1,9
5401        IL=8+I
5421        ITEM=3+I
5441        DO 70 J=1,N
5461        FID=DF2(IPH,J,I)
5481        CALL DTEST
5501     70 DF2(IPH,J,I)=FID
5521     80 N=NAD(IPH)
5541        IF(N)100,140,90
5561     90 IF(N-3)110,110,100
5581    100 IL=0
5601        CALL LIST
5621        IL=6
5641        CALL LIST
5661        NAD(IPH)=0
5681        IER=5
5701        CALL ERROR
5721        GO TO 140
5741    110 DO 130 I=10,12
5761        IL=9+I
5781        ITEM=3+I
5801        DO 130 J=1,N
5821        FID=DF2(IPH,J,I)
5841        CALL DTEST
5861    130 DF2(IPH,J,I)=FID
5881    140 RETURN
5901        END
```

TABLE 8 (Cont)

c. Subroutine NOYES

```
5921        SUBROUTINE NOYES
5941        COMMON IYEAR,ISWTCH(10)
5961        COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
5981       &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
6001       &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
6021       &ASH(25,3),AIH(25,3),AITR(25,3)
6041        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
6061       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
6081        COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
6101     10 I=1
6121        INPUT 700,NY
6141        IF(NO-NY)30,20,30
6161     20 NY=-1*I
6181        RETURN
6201     30 I=-1
6221        IF(NYES-NY)40,20,40
6241     40 IER=6
6261        CALL ERROR
6281        GO TO 10
6301    700 FORMAT(A1)
6321        END
```

TABLE 8 (Cont)

d. Subroutine DELETE

```
6341        SUBROUTINE DELETE
6361        COMMON IYEAR,ISWTCH(10)
6381        COMMON NAME(25,3),IAD(25,3,3),DF1(25,3),IDF1(25,2),
6401       &DF2(25,3,12)
6421        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
6441       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
6461        COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
6481        CALL EDIT1
6501        IF(N)10,10,20
6521    10  IER=2
6541        CALL ERROR
6561        GO TO 150
6581    20  M=NPH
6601        DO 140 I=1,M
6621        IPH=M+1-I
6641        DO 30 J=1,N,2
6661        IF(IPH-IDEL(J))30,40,30
6681    30  CONTINUE
6701        GO TO 140
6721    40  PRINT 700,IPH,(NAME(IPH,J),J=1,3)
6741        KILL=KILL+1
6761        KILLS(KILL)=IPH
6781        IF(IPH-NPH)50,100,100
6801    50  NPH1=NPH-1
6821        DO 90 K=IPH,NPH1
6841        KK=K+1
6861        DO 60 J=1,3
6881        NAME(K,J)=NAME(KK,J)
6901        DO 60 L=1,3
6921    60  IAD(K,J,L)=IAD(KK,J,L)
6941        DO 70 L=1,3
6961    70  DF1(K,L)=DF1(KK,L)
6981        DO 80 L=1,2
7001    80  IDF1(K,L)=IDF1(KK,L)
7021        DO 90 L=1,12
7041        DO 90 J=1,3
7061    90  DF2(K,J,L)=DF2(KK,J,L)
7081   100  NPH=NPH-1
7101   140  CONTINUE
7121        CALL PHASES
7141   150  RETURN
7161   700  FORMAT(13H DELETE PHASE,I3,1X,3A4)
7181        END
```

TABLE 8 (Cont)

e. Subroutine NEWPHA

```
7201        SUBROUTINE NEWPHA
7221        COMMON IYEAR,ISWTCH(10)
7241        COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
7261       &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
7281       &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
7301       &ASH(25,3),AIH(25,3),AITR(25,3)
7321        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
7341       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
7361        COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
7381        DO :0 I=1,6
7401        IL=I
7421        CALL UPDATE
7441    10  CONTINUE
7461        N=NAC(IPH)
7481        IF(N)40,40,20
7501    20  DO 30 I=7,17
7521        IL=I
7541        DO 30 J=1,N
7561        IP=J
7581        CALL UPDATE
7601    30  CONTINUE
7621    40  N=NAD(IPH)
7641        IF(N)70,70,50
7661    50  DO 60 I=18,21
7681        IL=I
7701        DO 60 J=1,N
7721        IP=J
7741        CALL UPDATE
7761    60  CONTINUE
7781    70  DO 80 I=1,22
7801        IL=I-1
7821        CALL LIST
7841    80  CONTINUE
7861        RETURN
7881        END
```

TABLE 8 (Cont)

f. Subroutine EDIT1

```
7901        SUBROUTINE EDIT1
7921        COMMON IYEAR,ISWTCH(10)
7941        COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
7961       &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
7981       &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
8001       &ASH(25,3),AIH(25,3),AITR(25,3)
8021        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
8041       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
8061        COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
8081        PRINT 700
8101     10 INPUT 701,IDEL
8121        IDEL(51)=0
8141        DO 80 I=1,25
8161        N=2*I-1
8181        IF(IDEL(N))30,90,20
8201     20 IF(IDEL(N)-NPH)50,50,30
8221     30 IER=6
8241     40 CALL ERROR
8261        GO TO 10
8281     50 IF(I-1)80,80,60
8301     60 IF(IDEL(N-1)-ICOMMA)70,80,70
8321     70 IER=1
8341        GO TO 40
8361     80 CONTINUE
8381     90 N=N-2
8401        RETURN
8421    700 FORMAT(" ENTER PHASE NUMBERS (XX,XX, . .)"/" TWO
8441       &DIGITS ARE REQUIRED FOR EACH PHASE"/)
8461    701 FORMAT(25(I2,A1),I2)
8481        END
```

TABLE 8 (Cont)

g. Subroutine LIST

```
8501        SUBROUTINE LIST
8521        COMMON IYEAR,ISWTCH(10)
8541        COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
8561       &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
8581       &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
8601       &ASH(25,3),AIH(25,3),AITR(25,3)
8621        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
8641       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
8661        COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
8681        NACC=NAC(IPH)
8701        NADD=NAD(IPH)
8721        IF(IL)100,100,110
8741   100  PRINT 700,IPH
8761        GO TO 200
8781   110  IF(IL-7)120,140,130
8801   120  GO TO (1,2,3,4,5,6),IL
8821     1  PRINT 701,(NAME(IPH,J),J=1,3)
8841        GO TO 200
8861     2  PRINT 702,ATP(IPH)
8881        GO TO 200
8901     3  PRINT 703,WK(IPH)
8921        GO TO 200
8941     4  PRINT 704, TOD(IPH)
8961        GO TO 200
8981     5  PRINT 705,NACC
9001        GO TO 200
9021     6  PRINT 706,NADD
9041        GO TO 200
9061   130  IF(IL-17)140,140,160
9081   140  K=IL-6
9101        IF(NACC)200,200,150
9121   150  GO TO (7,8,9,10,11,12,13,14,15,16,17),K
9141     7  PRINT 707,(NPLA(IPH,J),J=1,NACC)
9161        GO TO 200
9181     8  PRINT 708,(NFUEL(IPH,J),J=1,NACC)
9201        GO TO 200
9221     9  PRINT 709,(WX(IPH,J),J=1,NACC)
9241        GO TO 200
9261    10  PRINT 710,(GAS(IPH,J),J=1,NACC)
9281        GO TO 200
9301    11  PRINT 711,(AU(IPH,J),J=1,NACC)
9321        GO TO 200
9341    12  PRINT 712,(FU(IPH,J),J=1,NACC)
9361        GO TO 200
```

TABLE 8 (Cont)

g. Subroutine LIST (Cont)

```
9381    13 PRINT 713,(SFH(IPH,J),J=1,NACC)
9401       GO TO 200
9421    14 PRINT 714,(FIH(IPH,J),J=1,NACC)
9441       GO TO 200
9461    15 PRINT 715,(FTR(IPH,J),J=1,NACC)
9481       GO TO 200
9501    16 PRINT 716,(FSO(IPH,J),J=1,NACC)
9521       GO TO 200
9541    17 PRINT 717,(AMO(IPH,J),J=1,NACC)
9561       GO TO 200
9581   160 K=IL-17
9601       IF(NADD)200,200,170
9621   170 GO TO (18,19,20,21),K
9641    18 PRINT 718,(NACD(IPH,J),J=1,NADD)
9661       GO TO 200
9681    19 PRINT 719,(ASH(IPH,J),J=1,NADD)
9701       GO TO 200
9721    20 PRINT 720,(AIH(IPH,J),J=1,NADD)
9741       GO TO 200
9761    21 PRINT 721,(AITR(IPH,J),J=1,NADD)
9781   200 RETURN
9801   700 FORMAT(/29H DATA LIST FOR TRAINING PHASE,I3)
9821   701 FORMAT(15H 01 PHASE NAME ,3A4)
9841   702 FORMAT(19H 02 ATTRITION POINT,F7.4)
9861   703 FORMAT(18H 03 PHASE DURATION,F6.2,6H WEEKS)
9881   704 FORMAT(16H 04 TOUR OF DUTY,F6.2,7H MONTHS)
9901   705 FORMAT(21H 05 AIRCRAFT TYPES   ,I2)
9921   706 FORMAT(21H 06 INSTRUCTION TYPES,I2)
9941   707 FORMAT(24H 07 AIRCRAFT TYPES      ,3(1X,A4,2X))
9961   708 FORMAT(13H 08 FUEL TYPE,11X,3(1X,A4,2X))
9981   709 FORMAT(23H 09 FLYABLE WEATHER    ,3F7.3)
10001  710 FORMAT(22H 10 FUEL CONSUMPTION  ,3F7.2)
10021  711 FORMAT(22H 11 A/C UTILIZATION   ,3F7.2)
10041  712 FORMAT(22H 12 INSTRUCTOR UTIL.  ,3F7.2)
10061  713 FORMAT(17H 13 FLIGHT HOURS ,5X,3F7.2)
10081  714 FORMAT(22H 14 FLIGHT INST. HOURS,3F7.2)
10101  715 FORMAT(22H 15 INST. TR. PERIOD  ,3F7.2)
10121  716 FORMAT(22H 16 LSO RATIO         ,3F7.2)
10141  717 FORMAT(22H 17 MAINTAINENCE MEN  ,3F7.2)
10161  718 FORMAT(23H 18 ACADEMIC INSTRUCT. ,3(2X,A4,1X))
10181  719 FORMAT(17H 19 STUDENT HOURS,5X,3F7.2)
10201  720 FORMAT(22H 20 INSTRUCTOR HOURS  ,3F7.2)
10221  721 FORMAT(22H 21 INST. TR. PERIOD  ,3F7.2)
10241      END
```

TABLE 8 (Cont)

h. Subroutine UPDATE

```
10261       SUBROUTINE UPDATE
10281       COMMON SWITCH(11)
10301       COMMON NAME(25,3),IAD(25,3,3),DF1(25,3),IDF1(25,2),
10321      &DF2(25,3,12)
10341       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
10361      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
10381       COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
10401       IER=0
10421       IF(IL-1)20,20,130
10441    20 PRINT 700,IPH
10461       INPUT 701,(NAME(IPH,J),J=1,3)
10481       GO TO 500
10501   130 IF(IL-4)140,140,170
10521   140 K=IL-1
10541       PRINT 704,IL
10561       INPUT,FID
10581       ITEM=K
10601       DF1(IPH,K)=FID
10621       CALL DTEST
10641       DF1(IPH,K)=FID
10661       GO TO 500
10681   170 IF(IL-6)180,180,200
10701   180 K=IL-4
10721       PRINT 706,IL
10741       INPUT,IID
10761       IDF1(IPH,K)=IID
10781       ITEM=16
10801       CALL DTEST
10821       IDF1(IPH,K)=IID
10841       GO TO 500
10861   200 IF(IL-17)210,210,310
10881   210 N=IDF1(IPH,1)
10901       IF(IL-8)220,220,260
10921   220 K=IL-6
10941   230 IF(IP-N)250,250,500
10961   250 PRINT 707,IL,IP
10981       INPUT 701,IAD(IPH,IP,K)
11001       GO TO 500
11021   260 K=IL-8
11041       ITEM=IL-5
11061   270 IF(IP-N)280,280,500
11081   280 PRINT 709,IL,IP
11101       INPUT,FID
```

TABLE 8 (Cont)

h. Subroutine UPDATE (Cont)

```
11121       DF2(IPH,IP,K)=FID
11141       CALL DTEST
11161       DF2(IPH,IP,K)=FID
11181       GO TO 500
11201   310 IF(IL-21)320,320,500
11221   320 N=IDF1(IPH,2)
11241       IF(IL-18)340,330,340
11261   330 K=3
11281       GO TO 230
11301   340 K=IL-9
11321       ITEM=IL-6
11341       GO TO 270
11361   500 RETURN
11381   700 FORMAT(20H ENTER NAME OF PHASE,I3,15H (AAAAAAAAAAAA))
11401   701 FORMAT(3A4)
11421   702 FORMAT(41H ENTER FOLLOWING PHASE NUMBERS (XX,XX,XX))
11441   704 FORMAT(17H ENTER DATA FIELD,I3,10H (XXXX.XX))
11461   706 FORMAT(17H ENTER DATA FIELD,I3,4H (X))
11481   707 FORMAT(17H ENTER DATA FIELD,I3,1H-,I1,7H (AAAA))
11501   709 FORMAT(17H ENTER DATA FIELD,I3,1H-,I1,11H (XXX.XXXX))
11521       END
```

TABLE 8 (Cont)

i. Subroutine DTEST

```
11541       SUBROUTINE DTEST
11561       COMMON IYEAR,ISWTCH(10)
11581       COMMON IAD(25,3,4),DF1(25,3),IDF1(25,2),DF2(25,3,12)
11601       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
11621      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
11641       COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
11661       IF(ITEM-16)50,90,500
11681    50 IF(FID)70,500,60
11701    60 IF(FID-BMAX(ITEM))500,500,70
11721    70 K=IL
11741       IL=0
11761       CALL LIST
11781       IL=K
11801       CALL LIST
11821       PRINT 700,FID,BMAX(ITEM)
11841       CALL NOYES
11861       IF(NY)80,80,500
11881    80 PRINT 701
11901       INPUT,FID
11921       GO TO 50
11941    90 IF(IID)110,500,100
11961   100 IF(IID-3)500,500,110
11981   110 K=IL
12001       IL=0
12021       CALL LIST
12041       IL=K
12061       CALL LIST
12081       PRINT 703
12101       INPUT,IID
12121       GO TO 90
12141   500 RETURN
12161   700 FORMAT(11H DATA POINT,F9.4,23H EXCEEDS RANGE OF 0.0 -,F9.4/13
12181      &H ACCEPT (Y,N))
12201   701 FORMAT(31H ENTER CORRECT VALUE (XXX.XXXX))
12221   703 FORMAT(40H INVALID VALUE - ENTER CORRECT VALUE (X))
12241       END
```

TABLE 8 (Cont)

j. Subroutine PHASES

```
12261       SUBROUTINE PHASES
12281       COMMON IYEAR,ISWTCH(10)
12301       COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
12321      &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
12341      &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
12361      &ASH(25,3),AIH(25,3),AITR(25,3)
12381       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
12401      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
12421       COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
12441       PRINT 700
12461       IF(NPH)40,40,10
12481    10 DO 20 I=1,NPH
12501    20 PRINT 701,I,(NAME(I,J),J=1,3)
12521    30 PRINT 703
12541       RETURN
12561    40 PRINT 702
12581       GO TO 30
12601   700 FORMAT(//16H TRAINING PHASES/15H NO. PHASE NAME)
12621   701 FORMAT(I3,2X,3A4)
12641   702 FORMAT(10H NO PHASES/)
12661   703 FORMAT(//" ")
12681       END
```

TABLE 8 (Cont)

k. Subroutine ERROR

```
12701       SUBROUTINE ERROR
12721       COMMON IYEAR,ISWTCH(10)
12741       COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
12761      &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
12781      &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
12801      &ASH(25,3),AIH(25,3),AITR(25,3)
12821       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
12841      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
12861       COMMON IBC,IL,IP,N,ITEM,IDEL(51),BMAX(15)
12881       GO TO (2,3,4,5,6,7),IER
12901     2 PRINT 702
12921       GO TO 100
12941     3 PRINT 703
12961       GO TO 100
12981     4 PRINT 704
13001       GO TO 100
13021     5 PRINT 705
13041       GO TO 100
13061     6 PRINT 706
13081       GO TO 100
13101     7 PRINT 707
13121   100 RETURN
13141   702 FORMAT(21H COMMA MISSING REPEAT)
13161   703 FORMAT(30H PREVIOUS OPTION NOT PROCESSED)
13181   704 FORMAT(22H NO PHASES IN PIPELINE)
13201   705 FORMAT(22H 25 PHASES IN PIPELINE)
13221   706 FORMAT(37H MAX. FOR FIELD IS 3 - FIELD SET TO 0)
13241   707 FORMAT(22H INVALID DATA - REPEAT)
13261       END
```

IV. PROGRAM LSR2

PROGRAM DESCRIPTION

4.1 The purpose of PROGRAM LSR2 is to compute the student inputs, outputs, and attrites for each phase of pilot training. Upon entry, data file PIPE, which contains the phase sequence and attrition rates for each (student source) training pipeline, is opened and rewound. The user is then asked whether he wishes to have all training pipelines printed. A print switch, IPRT, is set to 0 with a No, "N," reply and to 1 for a Yes, "Y," reply.

4.2 If the level of complexity, LEVLSR, is 3 or 4, the user is asked whether the pipeline data, which may subsequently be modified, are to be saved on an output file, PIPES. With a Yes response, data file PIPES is opened and rewound. Subroutine PIPINP is called to enter the phase sequence and attrition rates for the next training pipeline from data file PIPE. Note that the first time Subroutine PIPINP is called, the first training pipeline is entered; the second time, the second pipeline is entered, etc.

4.3 When control returns, a test is made to determine whether an end of file on the data file has been encountered; i.e., the last training pipeline has been processed. If an end of file has not been reached, a transfer is made to Subroutine PIPENT to enter the appropriate pilot training rates. Student inputs, outputs, and attrites for each phase of training are then printed. If the user has specified that the training pipeline data are to be saved, all data for the training pipeline which are currently in memory are written on data file PIPES. Subroutine PIPINP is then called to enter the next pipeline.

4.4 When an end of file is encountered, the user is permitted to add a new training pipeline if the level of complexity LEVLSR is 3 or 4. The new

pipeline is developed in Subroutine MPIPE. When all pipelines have been entered and their student statistics computed, data files PIPE and PIPES are closed. Composite student statistics (an aggregation over all training pipelines) are printed and control passes to PROGRAM LSR3.

SUBROUTINE MPIPE

4.5 Subroutine MPIPE modifies the pipeline data when the level of complexity LEVLSR is 3 or 4. Upon entry, a test is made to determine whether any training phases exist in the pipeline. If no phases exist, NPHP = 0, the user must generate a new pipeline by adding training phases. When some training phases exist, Subroutine MPIPE prints the name of the pipeline, e.g., naval officers, and asks the user whether he wishes to make any modifications to the pipeline. If no modifications are to be made, Subroutine PIPER is called to check the pipeline for consistency. If Subroutine PIPER has deleted all training phases, Subroutine MPIPE recycles to enter all pipeline information; otherwise, a return is made to the calling routine.

4.6 When the user specifies modifications to the training pipeline, the user is asked whether any training phases are to be deleted from the training pipeline. With a Yes command, the number of the phase to be deleted is entered, and Subroutine DPIPE is called to remove the specified training phase. This process is continued until the user has deleted the required number of training phases.

4.7 The user is next asked whether he desires to add a new training phase. If the reply is Yes, the user enters the number of the new training phase, the following phase numbers, and the phase attrition rate. The new phase number is checked for validity. New phases are added until the user terminates this process.

4.8 The user is asked whether he wishes a list of the phase sequences and attrition rates of the training pipeline. With a Yes reply, Subroutine PIPRT is entered to print the pipeline data.

4.9 The user is asked whether modifications are to be made to the pipeline. When modifications are to occur, the user enters the number of the phase to be modified, IPH, and a switch,ISW, indicating whether he wishes to revise the following phases, ISW = 0, or the attrition rate, ISW = 1. The user enters the appropriate value(s) and continues to modify other phases. When the user terminates the modification process, the subroutine returns to the beginning and continues until the user terminates the modification process.

SUBROUTINE PIPRT

4.10 Subroutine PIPRT prints the sequence of phases within the training pipeline and their respective attrition rates. Upon entry, a loop is developed to print all training phase data. When the printing process is completed, control is returned to the calling routine.

SUBROUTINE LOADSO

4.11 Subroutine LOADSO develops an arbitrary pilot training rate of 1,000 students for each terminal phase in the training pipeline. Upon entry, the terminal pipeline phases are selected and assigned an arbitrary PTR of 1,000 student pilots. Subroutine OUTPUT is called to develop the PTR for the remaining training phases. If the PTR of all training phases cannot be computed, Subroutine LOADSO deletes all phases in the pipeline, prints a message to the user indicating that the pipeline has been purged of its training phases, and returns to the calling routine. This check is made since it is possible for the user to generate a training pipeline with an invalid sequence of training phases (e.g., phase 1 goes to phase 2 and phase 2 returns to phase 1).

SUBROUTINE PIPER

4.12 Subroutine PIPER checks for obvious inconsistencies in the training pipeline. Upon entry, each traning phase in the pipeline is checked to determine whether a duplicate phase occurs. For example, advanced Helo may appear twice in the training pipeline. Should duplicate phases occur, the duplicate phase is deleted through Subroutine DPIPE. A second test is made to determine the number of phases that are immediate precursors to a particular training phase, i.e., how many training phases lead directly to Basic Prop. Should the number of training phases directly leading to any phase exceed one, an error message is listed and the user makes the appropriate modifications. The attrition rate for each training phase is then checked to determine whether it exceeds 100 percent. Should this occur, the user must make the appropriate modifications. Control then returns to the calling routine.

SUBROUTINE NOYES

4.13 Subroutine NOYES reads a Yes, "Y," or No, "N," response from the time-sharing terminal. Switch NY, which is in common, is set to -1 for a No response and to 1 for a Yes response. When a Yes or No reply is required in PROGRAM LSR2 or its subroutines, Subroutine NOYES is called.

SUBROUTINE ERROR

4.14 Subroutine ERROR prints diagnostic messages. In accordance with an error type switch, IER, diagnostic message is displayed.

SUBROUTINE PIPENT

4.15 The purpose of Subroutine PIPENT is to enter the PTR for phases in the training pipeline. Upon entry, the PTR for all training phases is set to zero. Next the user enters the PTR for particular phases in the training pipeline. A

validity check is made to determine whether the user-supplied phase number exists in the training pipeline.

4.16 When the user has completed entering the appropriate PTR, Subroutine OUTPUT is called to develop the PTR for the remaining training phases. Should Subroutine OUTPUT be unable to compute the PTR of the remaining phases, all PTRs are set to zero, an error message is displayed, and the user is requested to re-enter the PTR data.

4.17 Subroutine SMOOTH is called to reduce phase PTRs when appropriate. For example, Basic Jet A could lead solely to Basic Jet B with respective PTRs of 10,000 and 1,000. Assuming a zero attrition rate, Subroutine SMOOTH would reduce the PTR of Basic Jet A to 1,000 since that is the minimum number of students required to satisfy the PTR of Basic Jet B. Control returns to the calling routine.

SUBROUTINE PIPINP

4.18 Subroutine PIPINP enters the pipeline data into memory. Upon entry, data file PIPE is read to determine the name of the training pipeline and the number of phases that exist in the training pipeline. Should a negative number of training phases exist in the pipeline, end of file is noted and control returned. Should no phases exist, the user is asked whether he wishes to add phases. With a Yes reply, Subroutine MPIPE is entered to add training phases, and Subroutine PIPER and LOADSO are called to check for a consistent phase sequence.

4.19 Should the user decline to add training phases, control is returned to the calling routine. When training phases exist in the training pipeline, all pipeline data are read from data file PIPE. When all data are read, a test is made to determine if any training phases have been deleted in PROGRAM LSR1. If phases have been deleted previously, the deleted phases—when applicable—are purged from the pipeline. Variable KILLS records the numbers of the phases which were deleted in PROGRAM LSR1.

4.20 Subroutine PIPRT is called if the user desired to have the pipeline data listed (IPRT = 1) and Subroutine MPIPE called if the user requested to modify the pipeline data (LEVLSR = 3 or 4). Subroutines PIPER and LOADSO are called to check for consistency in the pipeline data. Control is returned to PROGRAM LSR2.

SUBROUTINE DPIPE

4.21 Subroutine DPIPE deletes phases from the training pipeline. Upon entry, the phase number to be deleted is checked against the phase numbers in the pipeline. When the appropriate phase is found, it is removed from the pipeline and all pipeline data are revised. For example, should a phase Primary

Training be the precursor phase to Basic Jet A and Basic Prop, with Basic Prop being deleted, the revised pipeline would portray Primary Training leading solely to Basic Jet A. Control is returned to the calling routine.

SUBROUTINE OUTPUT

4.22 Subroutine OUTPUT computes the student output for the phases of training. Upon entry, the training phases are scanned in reverse sequence until a phase is found with a PTR, i.e., Advanced Jet. The PTRs of the phases leading to this phase are selected and their appropriate PTR computed. This process is continued until the PTRs are computed for all training phases or until no more computations can be made. When all PTRs are computed, Subroutine OUTPUT returns to the calling routine.

4.23 However, since the user may enter the PTR of beginning phases, i.e., Primary Training, rather than the terminal phases, i.e., Advanced Helicopter, Subroutine OUTPUT may not be able to compute all PTRs. Note that Subroutine OUTPUT only computes PTRs in a reverse sequence; i.e., the PTR of Basic Jet A is the PTR of Basic Jet B divided by 1 minus the Basic Jet B attrition rate. Subroutine OUTFOR is called, when necessary, to compute the PTR utilizing a forward progression through the training pipeline. Should neither Subroutine OUTPUT or Subroutine OUTFOR be able to compute the PTR for all training phases, an error condition is established and control returned.

SUBROUTINE OUTFOR

4.24 Subroutine OUTFOR computes the training phase PTRs by utilizing a forward movement through the training pipeline. Upon entry, the training phases are sequentially scanned to find a phase with a PTR. The PTR of the following phase, when applicable, is computed. Should a branch exist in the pipeline, e.g., Primary Training leading to Basic Prop and Basic Jet, with the PTR of the first phase (Primary Training) known, the user is requested to distribute the output of the first phase (Primary Training) to the other phases (Basic Jet and Basic Prop). The process continues until no further computations can be performed. Control is then returned to Subroutine OUTPUT.

SUBROUTINE SMOOTH

4.25 Subroutine SMOOTH scans the pipeline and minimizes the phase PTRs when possible. The programmer should note the similarity between Subroutine SMOOTH and Subroutines OUTPUT and OUTFOR. The difference in these routines is that all phase PTRs have been computed when Subroutine SMOOTH is entered.

4.26 Upon entry, all phase PTRs are scanned and recomputed. When the computed phase PTR is less than the actual phase PTR, the phase PTR is replaced by the computed value. For example, Basic Jet A leads solely to Basic Jet B and the

respective PTRs are 10,000 and 1,000 students. Should Basic Jet B have no attrition, only 1,000 students need to be annually trained in Basic Jet A. Consequently, the PTR of Basic Jet A would be reduced accordingly.

4.27 A flow chart of PROGRAM LSR2 is shown in Figure 4. Table 9 contains the variable dictionary of PROGRAM LSR2; the program and subroutine dictionary is provided in Table 10. The program listing is shown in Table 11.

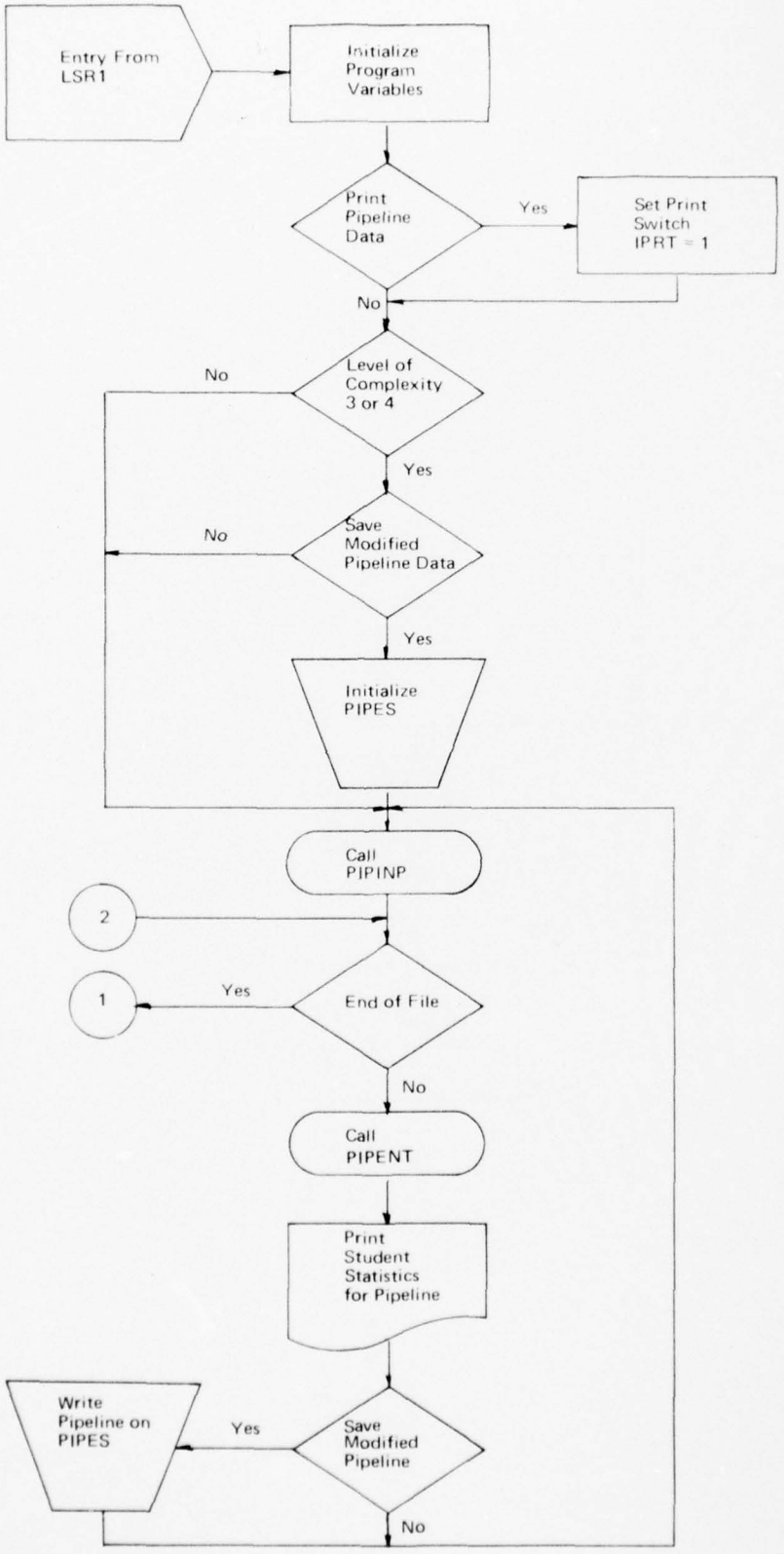


FIGURE 4. PROGRAM LSR2 FLOW CHART

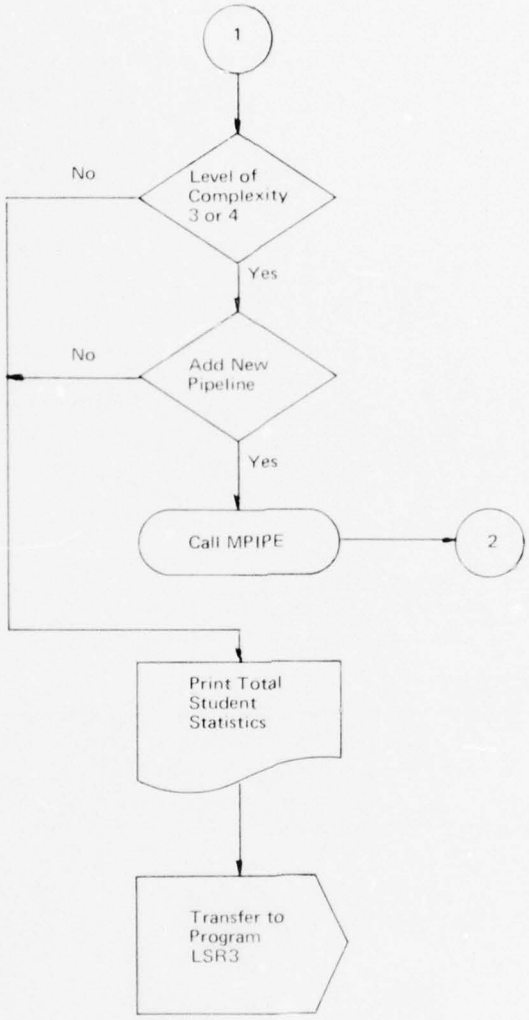


FIGURE 4 (Cont)

a. Subroutine MPIPE

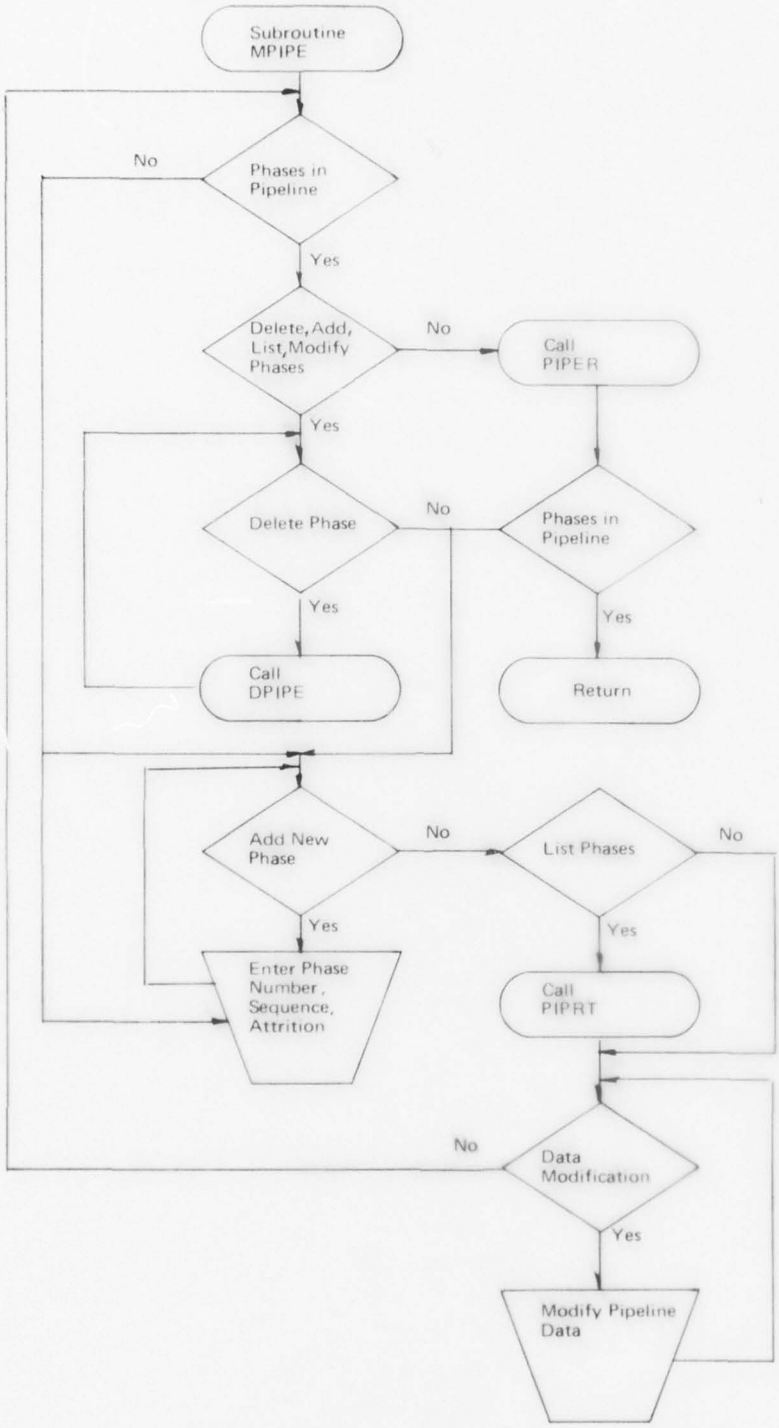
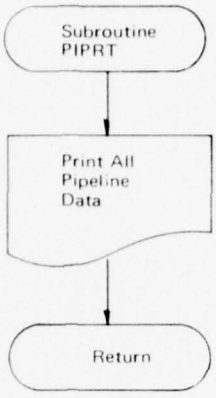


FIGURE 4 (Cont)

b. Subroutine PIPRT



c. Subroutine LOADSO

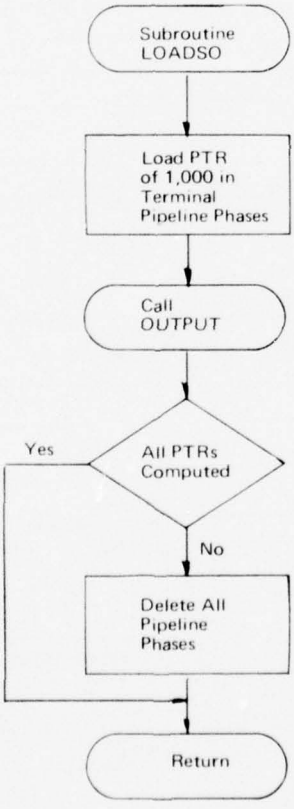


FIGURE 4 (Cont)

d. Subroutine PIPER

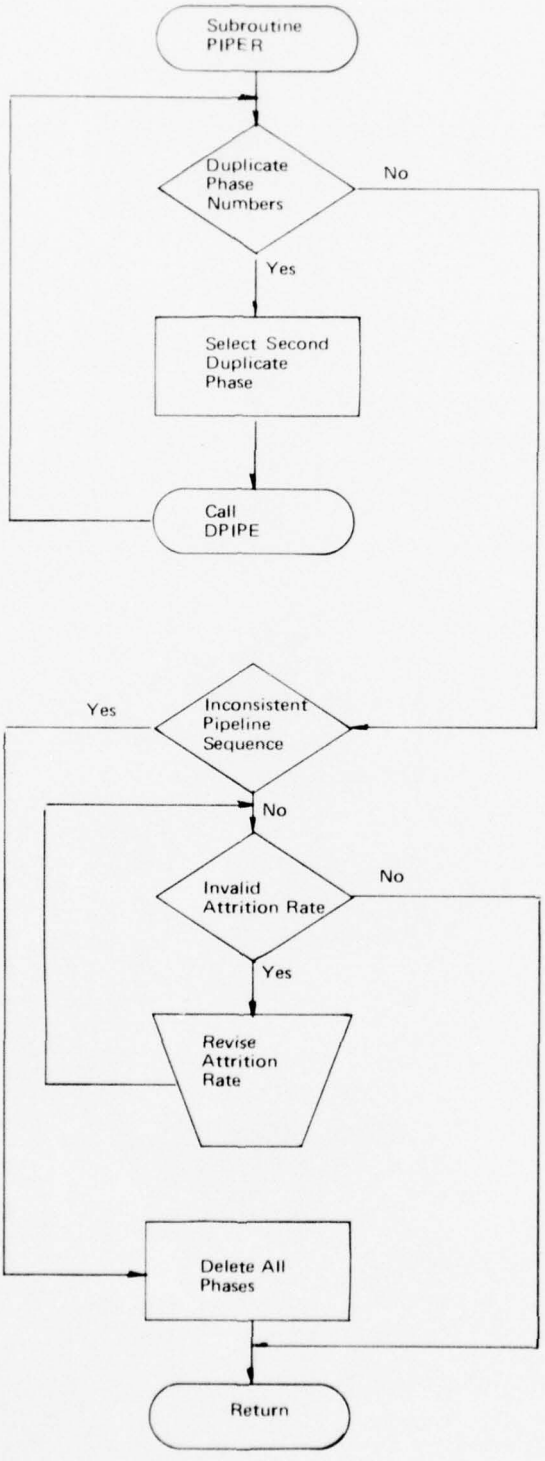


FIGURE 4 (Cont)

e. Subroutine NOYES

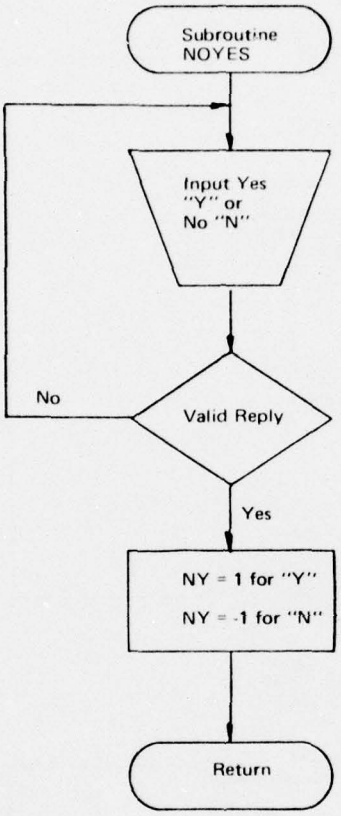


FIGURE 4 (Cont)

f. Subroutine ERROR

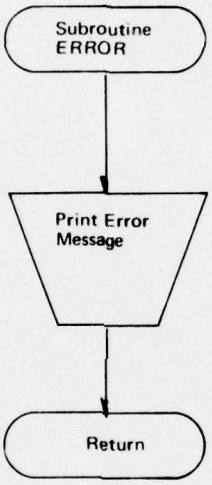


FIGURE 4 (Cont)

g. Subroutine PIPENT

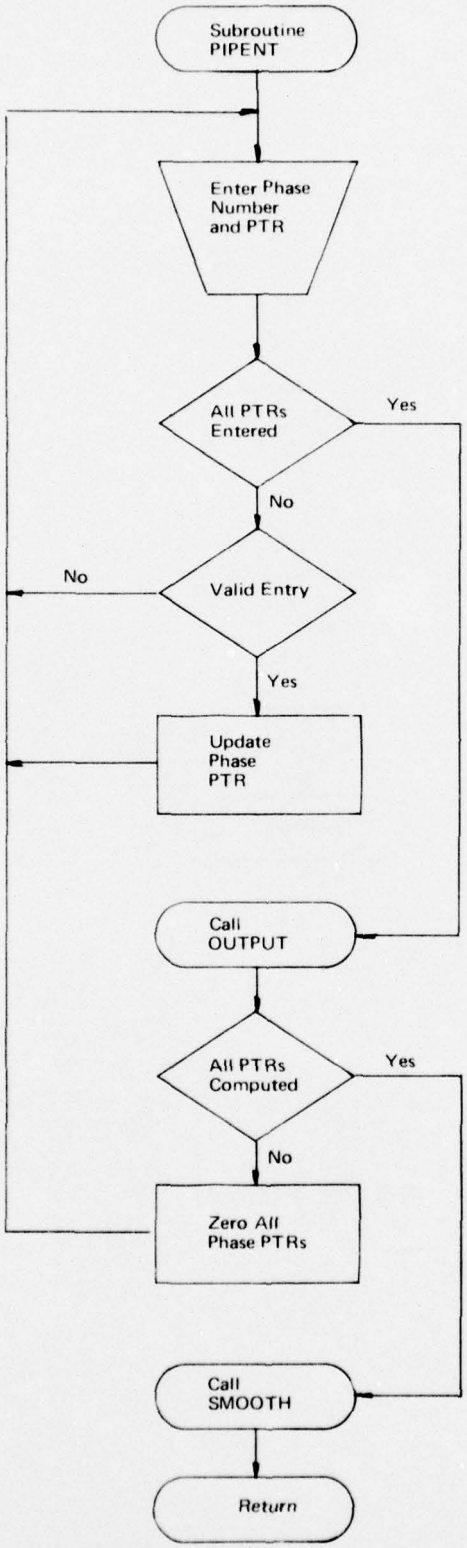


FIGURE 4 (Cont)

h. Subroutine PIPINP

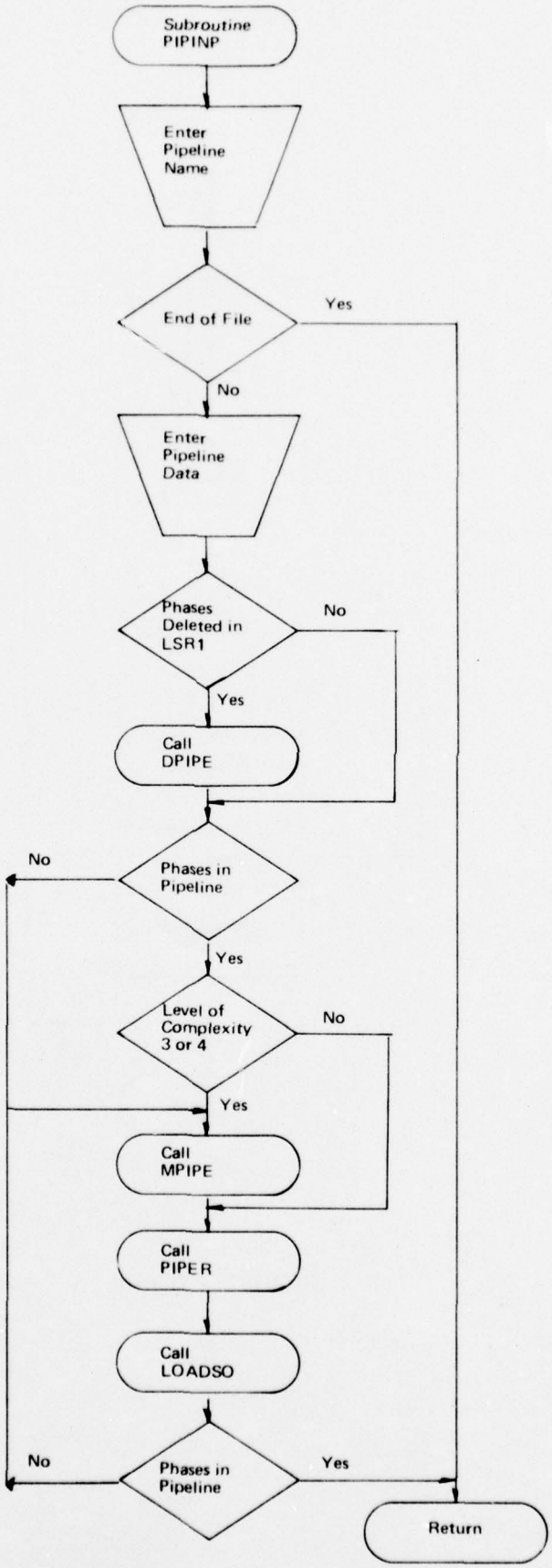


FIGURE 4 (Cont)

i. Subroutine DPIPE

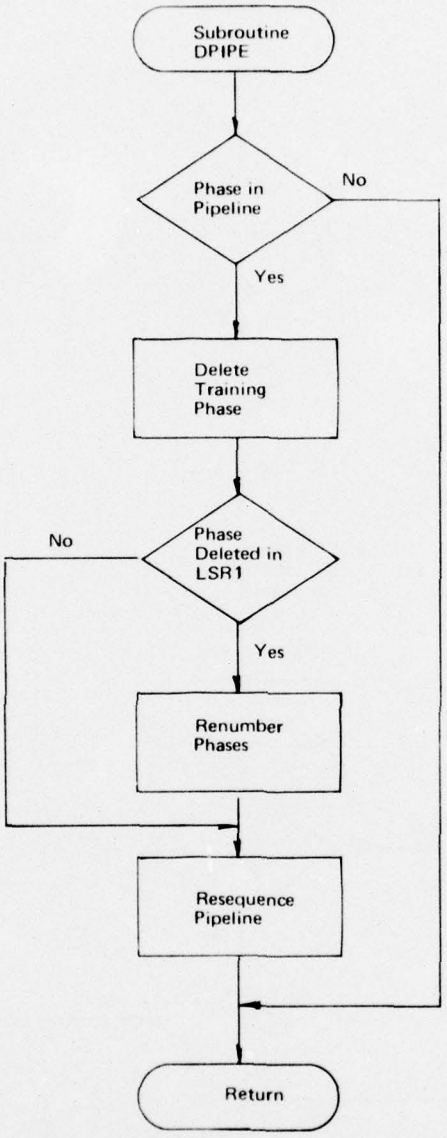


FIGURE 4 (Cont)

j. Subroutine OUTPUT

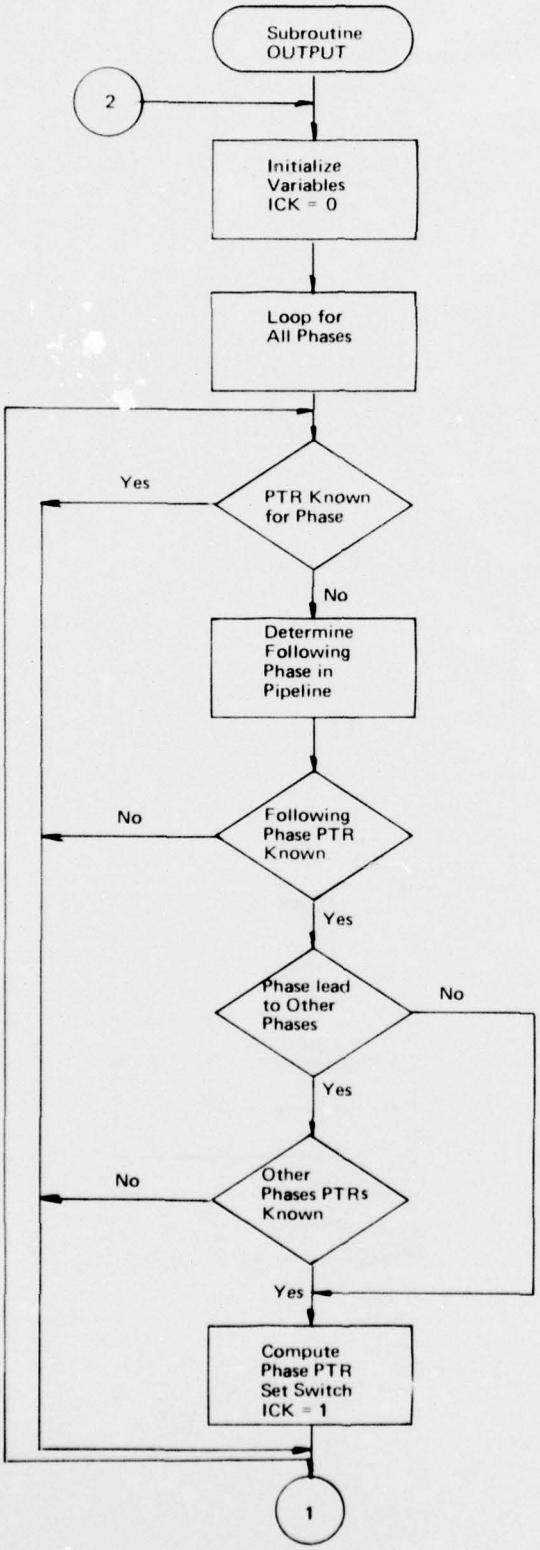


FIGURE 4 (Cont)

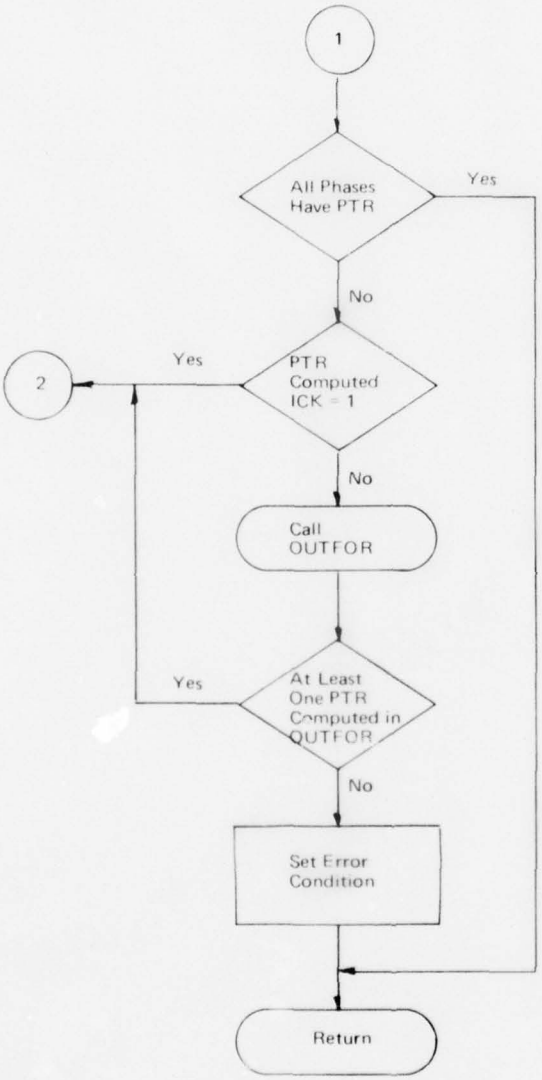


FIGURE 4 (Cont)

k. Subroutine OUTFOR

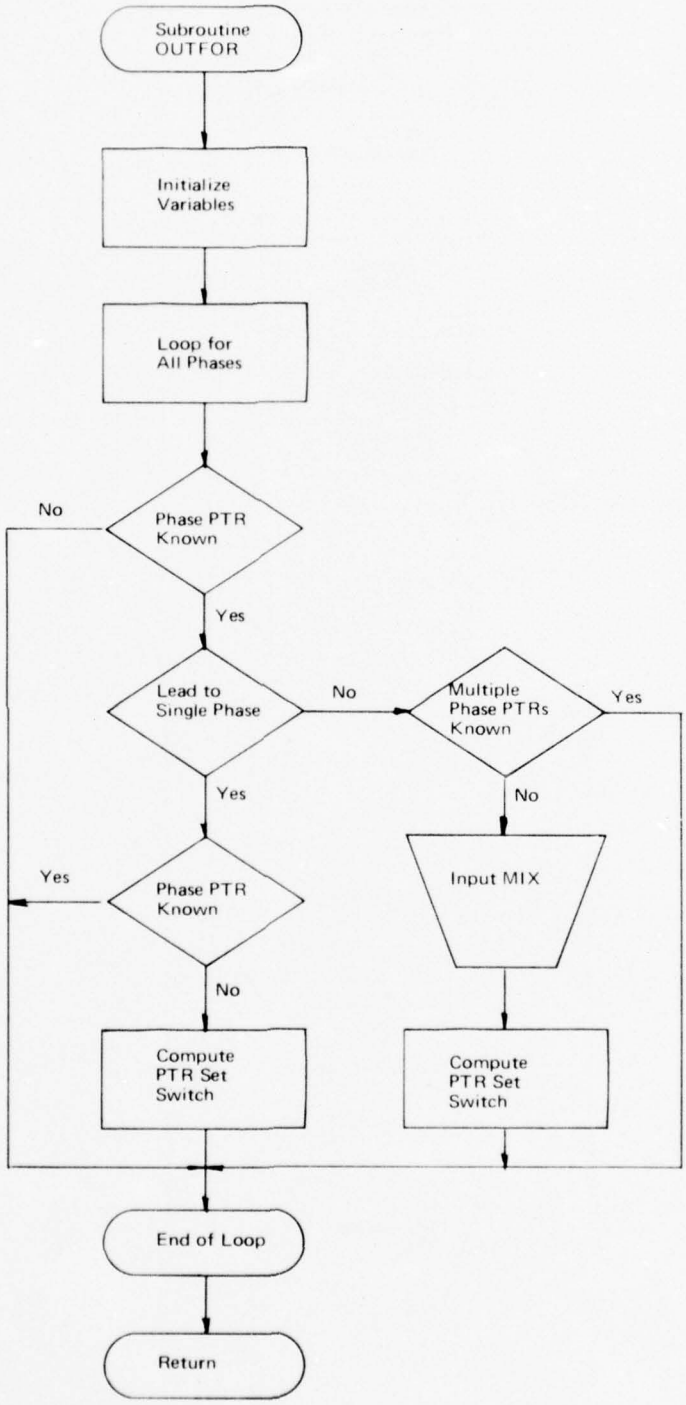


FIGURE 4 (Cont)

1. Subroutine SMOOTH

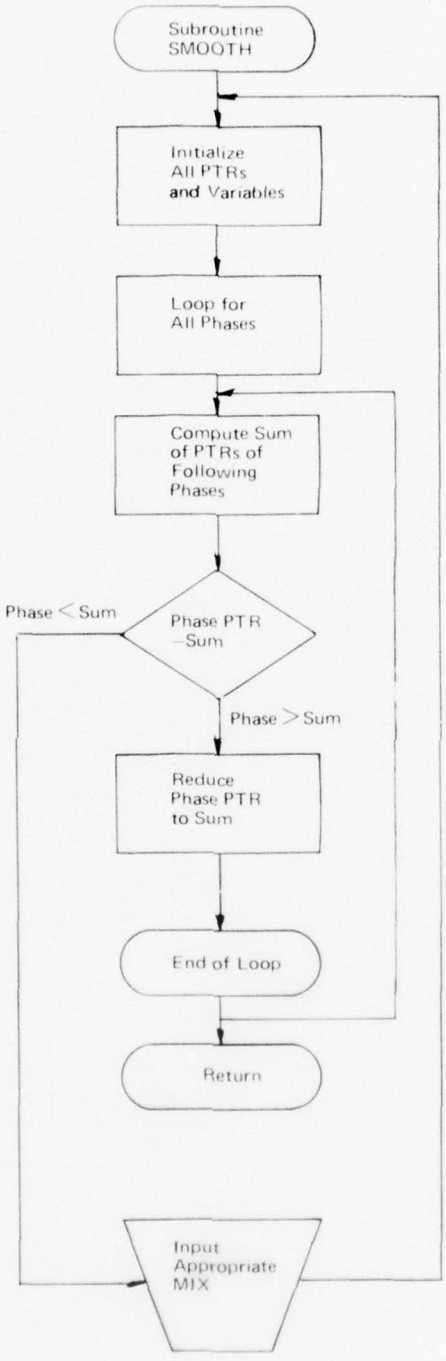


FIGURE 4 (Cont)

TABLE 9

PROGRAM LSR2 VARIABLE DICTIONARY*

Location	Variable Name	Dimension	Type	Description
Common	NPHP	1	I	Number of training phases in training pipeline
Common	IPHASE	25,4	I	Phase sequence for Ith item in pipeline. IPHASE (I, J), J = 1, 3 are the following phase numbers of phase IPHASE (I, 4)
Common	ATR	25	F	Attrition rate for phase I
Common	PNAME	3	A	Name of training pipeline
Common	IPRT	1	I	Print switch IPTR = 0: do not print pipeline data = 1: print pipeline data
Common	NPSW	1	I	End of file switch NPSW < end of file reached in data file PIPE
Common	LSOSW	1	I	Switch used to check validity of pipeline
Common	IDATA	5	I	Following phase numbers. Note even values are "g"
Common	IC	1	I	Temporary switch
Common	ISAVE	1	I	Save modified pipeline data ISAVE = 0: do not save pipeline data = 1: save pipeline data
Common	ATL	1	F	Number of student attrites
Common	SIN	1	F	Student input
Common	LI	1	I	Line number on output file
Common	SOUT	1	F	PTR for a particular phase

* Note that the first 1,450 words of common are listed in Table 3.

TABLE 10

PROGRAM LSR2 PROGRAM AND SUBROUTINE DICTIONARY

LSR2	Develops linkage to sequentially input all data associated with each pilot training pipeline. Prints student statistics for each pipeline.
MPIPE	Modifies pipeline data by adding and deleting training phases and revising particular data points.
PIPRT	Prints the data associated with a particular training pipeline.
LOADSO	Loads a PTR of 1,000 students into each terminal pipeline phase for validation purposes.
PIPER	Tests a particular pipeline for logical inconsistencies.
NOYES	Reads a No, "N," or Yes, "Y," response from the time-sharing terminal.
ERROR	Prints error messages when invalid data are entered by the user.
PIPENT	Reads from the time-sharing terminal the user-supplied pilot training rates.
PIPINP	Enters pipeline data.
DPIPE	Deletes training phases from the training pipeline.
OUTPUT	Develops the PTR for phases using a reverse scan of the training pipeline.
OUTFOR	Develops the PTR for training phases using a forward scan of the training pipeline.
SMOOTH	Revises the phase PTRs so that overproduction will not occur in any phase.

TABLE 11

PROGRAM LSR2 LISTING

```
102       COMMON SWITCH(11),NAME(25,3),SPACE(25,50)
122       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
142      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
162       COMMON NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSOSW
182      &,IDATA(5),IC
202       OPENFILE "PIPE"
222       IDATA(2)=ICOMMA
242       IDATA(4)=ICOMMA
262       LSOSW=0
282       REWIND "PIPE"
302       DO 10 I=1,NPH
322       TSOUT(I)=0.0
342    10 SI(I)=0.0
362       PRINT 700
382       CALL NOYES
402       IPRT=NY
422       ISAVE=0
442       IF(LEVLSR-3)40,20,20
462    20 PRINT 708
482       CALL NOYES
502       IF(NY)40,40,30
522    30 ISAVE=1
542       OPENFILE "PIPES"
562       REWIND "PIPES"
582       LI=1000
602    40 NPSW=1
622   100 CALL PIPINP
642       IF(NPSW)200,100,110
662   110 CALL PIPENT
682       PRINT 701,PNAME
702       DO 120 I=1,NPHP
722       K=IPHASE(I,4)
742       SIN=SO(K)/(1.0-ATR(K))
762       ATL=SIN-SO(K)
782       TSOUT(K)=TSOUT(K)+SO(K)
802       SI(K)=SI(K)+SIN
822   120 PRINT 702,(NAME(K,J),J=1,3),SIN,SO(K),ATL
842       PRINT 707
862       IF(ISAVE)100,100,130
```

TABLE 11(Cont)

```
882  130 WRITE("PIPES",709)LI,NPHP,PNAME
902      LI=LI+5
922      DØ 140 I=1,NPHP
942      K=IPHASE(I,4)
962      WRITE("PIPES",710)LI,(IPHASE(I,J),J=1,4),ATR(K)
982      KILL=0
1002 140 LI=LI+5
1022     GØ TØ 100
1042 200 IF(LEVLSR-2)300,300,210
1062 210 PRINT 703
1082     CALL NØYES
1102     IF(NY)300,300,220
1122 220 PRINT 704
1142     INPUT 705,PNAME
1162     NPHP=0
1182     CALL MPIPE
1202     IF(NPHP)210,210,110
1222 300 CLØSEFILE "PIPE"
1242     IF(ISAVE)320,320,310
1262 310 WRITE("PIPES",709)LI,NPSW,PNAME
1282     CLØSEFILE "PIPES"
1302 320 PRINT 706
1322     DØ 400 I=1,NPH
1342     ATL=SI(I)-TSØUT(I)
1362 400 PRINT 702,(NAME(I,J),J=1,3),SI(I),TSØUT(I),ATL
1382     CHAIN "XLSR3*"
1402 700 FØRMAT(26H PRINT ALL PIPELINES (Y,N))
1422 701 FØRMAT(//5X,14HSTUDENT TYPE  ,3A4//18X,22H.STUDENT   ST
1442    &ATISTICS./40H TRAINING PHASE   INPUT  ØUTPUT ATTRITES/)
1462 702 FØRMAT(1X,3A4,F10.0,2F8.0)
1482 703 FØRMAT(25H ADD A NEW PIPELINE (Y,N))
1502 704 FØRMAT(38H ENTER NAME ØF PIPELINE (AAAAAAAAAAAA))
1522 705 FØRMAT(3A4)
1542 706 FØRMAT(//5X,"TØTAL FØR ALL STUDENT TYPES"//18X,22H.STU
1562    &DENT   STATISTICS./" TRAINING PHASE   INPUT  ØUTPUT ATT
1582    &RITES"/)
1602 707 FØRMAT(//" ")
1622 708 FØRMAT(" SAVE MØDIFIED PIPELINES (Y,N)")
1642 709 FØRMAT(2I4,3A4)
1662 710 FØRMAT(I4,1X,4I3,F8.4)
1682     END
```

TABLE 11 (Cont)

a. Subroutine MPIPE

```
1702       SUBR0UTINE MPIPE
1722       C0MM0N SWITCH(11),NAME(25,3),SPACE(25,50)
1742       C0MM0N IC0MMA,IBLANK,N0,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
1762      &AFD,KILL,IID,FID,KILLS(25),SI(25),TS0UT(25),S0(25)
1782       C0MM0N NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LS0SW
1802      &,IDATA(5),IC
1822    10 IF(NPHP)100,100,20
1842    20 PRINT 700,PNAME
1862       CALL N0YES
1882       IF(NY)500,500,30
1902    30 PRINT 701
1922       CALL N0YES
1942       IF(NY)50,50,40
1962    40 PRINT 711
1982       IID=1
2002    45 INPUT, IPH
2022       IF(IPH)50,50,46
2042    46 CALL DPIPE
2062       PRINT 710
2082       G0 T0 45
2102    50 PRINT 702
2122       CALL N0YES
2142       IF(NY)60,60,70
2162    60 IF(NPHP)90,90,200
2182    70 IF(NPHP-NPH)110,80,80
2202    80 IER=5
2222       CALL ERR0R
2242       G0 T0 30
2262    90 IER=4
2282       CALL ERR0R
2302   100 NPHP=0
2322   110 N=NPHP
2342       NPHP=NPHP+1
2362       PRINT 703
2382   120 INPUT,IPH
2402       IF(IPH)130,130,140
2422   130 IER=7
2442       CALL ERR0R
2462       G0 T0 120
```

TABLE 11 (Cont)

a. Subroutine MPIPE (Cont)

```
2482   140 IF(IPH-NPH)150,150,130
2502   150 IF(N)190,190,170
2522   160 IER=1
2542       CALL ERRØR
2562       GØ TØ 50
2582   170 DØ 180 I=1,N
2602       IF(IPHASE(I,4)-IPH)180,160,180
2622   180 CØNTINUE
2642   190 PRINT 704
2662       INPUT,(IPHASE(NPHP,J),J=1,3),ATR(IPH)
2682       IPHASE(NPHP,4)=IPH
2702       GØ TØ 50
2722   200 PRINT 705
2742       CALL NØYES
2762       IF(NY)220,220,210
2782   210 CALL PIPRT
2802   220 PRINT 706
2822       CALL NØYES
2842       IF(NY)10,10,230
2862   230 PRINT 707
2882   240 INPUT,IPH,ISW
2902       IF(IPH)270,10,245
2922   245 N=0
2942       DØ 260 I=1,NPHP
2962       IF(IPHASE(I,4)-IPH)260,250,260
2982   250 N=I
3002       GØ TØ 280
3022   260 CØNTINUE
3042   270 IER=2
3062       CALL ERRØR
3082       CALL PIPRT
3102       GØ TØ 240
3122   280 IF(ISW)270,290,310
3142   290 PRINT 708
3162       INPUT,(IPHASE(N,J),J=1,3)
3182   300 PRINT 710
3202       GØ TØ 240
3222   310 PRINT 709
3242       INPUT,ATR(IPH)
3262       GØ TØ 300
3282   500 CALL PIPER
3302       IF(NPHP)90,90,510
3322   510 RETURN
```

TABLE 11

a. Subroutine MPIPE (Cont)

```
3342   700 FØRMAT(//" PIPELINE ",3A4,/" ANY DELETIØNS, ADDITIØNS, LI
3362      &STS ØR MØDIFICATIØNS (Y,N)")
3382   701 FØRMAT(24H DELETE ANY PHASES (Y,N))
3402   702 FØRMAT(" ADD A NEW PHASE (Y,N)")
3422   703 FØRMAT(31H ENTER NUMBER ØF NEW PHASE (XX))
3442   704 FØRMAT(42H ENTER FØLLØWING PHASES AND ATTRITIØN RATE/
3462      &48H (XX,XX,XX,.XXX) ALL DATA FIELDS MUST BE ENTERED/)
3482   705 FØRMAT(25H LIST PIPELINE DATA (Y,N))
3502   706 FØRMAT(21H MØDIFY A PHASE (Y,N))
3522   707 FØRMAT(" ENTER PHASE NUMBER AND SWITCH (XX,X)"/" SWITCH =
3542     & 0 - MØDIFY FØLLØWING PHASES"/"          = 1 - MØDIFY ATTRIT
3562      &IØN RATE"/" PHASE  = 0,0 IMPLIES NØ FURTHER MØDIFICATIØNS")
3582   708 FØRMAT(34H ENTER FØLLØWING PHASES (XX,XX,XX))
3602   709 FØRMAT(28H ENTER ATTRITIØN RATE (.XXX))
3622   710 FØRMAT(5H NEXT)
3642   711 FØRMAT(" ENTER PHASE NUMBERS (XX)"/" ENTER 0, FØR NØ FU
3662      &RTHER DELETIØNS")
3682       END
```

TABLE 11 (Cont)

b. Subroutine PIPRT

```
3702       SUBRØUTINE PIPRT
3722       CØMMØN SWITCH(11),NAME(25,3),SPACE(25,50)
3742       CØMMØN ICØMMA,IBLANK,NØ,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
3762      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSØUT(25),SØ(25)
3782       CØMMØN NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSØSW
3802      &,IDATA(5),IC
3822       PRINT 700,PNAME
3842       IF(NPHP)80,80,10
3862    10 DØ 60 K=1,NPHP
3882       I=IPHASE(K,4)
3902       IC=1
3922       DØ 30 J=1,3
3942       IF(IPHASE(K,J))30,30,20
3962    20 IDATA(IC)=IPHASE(K,J)
3982       IC=IC+2
4002    30 CØNTINUE
4022       IC=IC-2
4042       IF(IC)50,50,40
4062    40 PRINT 701,I,(NAME(I,J),J=1,3),ATR(I),(IDATA(J),J=1,IC)
4082       GØ TØ 60
4102    50 PRINT 701,I,(NAME(I,J),J=1,3),ATR(I)
4122    60 CØNTINUE
4142    70 PRINT 703
4162       RETURN
4182    80 PRINT 702
4202       GØ TØ 70
4222   700 FØRMAT(//27H  TRAINING  PIPELINE  FØR  ,3A4//6H PHASE,
4242      &13X,20HATTRITIØN  FØLLØWING/37H  NØ.   PHASE NAME    RATE
4262      &     PHASES/)
4282   701 FØRMAT(I4,4X,3A4,F7.4,I7,2(A1,I2))
4302   702 FØRMAT(20H   NØ CURRENT PHASES)
4322   703 FØRMAT(1X)
4342       END
```

TABLE 11 (Cont)

c. Subroutine LOADSO

```
4362        SUBR0UTINE L0ADS0
4382        C0MM0N SWITCH(11),NAME(25,3),SPACE(25,50)
4402        C0MM0N IC0MMA,IBLANK,N0,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
4422       &AFD,KILL,IID,FID,KILLS(25),SI(25),TS0UT(25),S0(25)
4442        C0MM0N NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LS0SW
4462       &,IDATA(5),IC
4482        IF(NPHP)40,40,10
4502    10  D0 30 K=1,NPHP
4522        I=IPHASE(K,4)
4542        S0(I)=0.0
4562        D0 20 J=1,3
4582        IF(IPHASE(K,J))30,20,30
4602    20  C0NTINUE
4622        S0(I)=-1000.
4642    30  C0NTINUE
4662        LS0SW=1
4682        CALL 0UTPUT
4702        LS0SW=0
4722        IF(IER)40,40,50
4742    40  RETURN
4762    50  IER=3
4782        CALL ERR0R
4802        NPHP=0
4822        G0 T0 40
4842        END
```

TABLE 11 (Cont)

d. Subroutine PIPER

```
4862        SUBRØUTINE PIPER
4882        CØMMØN SWITCH(11),NAME(25,3),SPACE(25,50)
4902        CØMMØN ICØMMA,IBLANK,NØ,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
4922       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSØUT(25),SØ(25)
4942        CØMMØN NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSØSW
4962       &,IDATA(5),IC
4982     10 IF(NPHP)170,170,20
5002     20 DØ 140 I=1,NPHP
5022        IPH=IPHASE(I,4)
5042        IF(IPH)40,40,30
5062     30 IF(IPH-NPH)50,50,40
5082     40 PRINT 700,IPH
5102        CALL DPIPE
5122        GØ TØ 10
5142     50 DØ 90 J=1,3
5162        IF(IPHASE(I,J))80,90,60
5182     60 IF(IPHASE(I,J)-IPH)70,80,70
5202     70 IF(IPHASE(I,J)-NPH)90,90,80
5222     80 PRINT 701,IPH,(IPHASE(I,K),K=1,3)
5242        INPUT,(IPHASE(I,K),K=1,3)
5262        GØ TØ 20
5282     90 CØNTINUE
5302        IC=0
5322        DØ 130 K=1,NPHP
5342        IF(I-K)100,110,100
5362    100 IF(IPHASE(K,4)-IPH)110,40,110
5382    110 DØ 130 J=1,3
5402        IF(IPHASE(K,J)-IPH)130,120,130
5422    120 IC=IC+1
5442    130 CØNTINUE
```

TABLE 11 (Cont)

d. Subroutine PIPER (Cont)

```
5462        IF(IC-1)135,135,160
5482   135  IF(ATR(IPH))137,140,136
5502   136  IF(ATR(IPH)-1.0)140,137,137
5522   137  PRINT 702,IPH,ATR(IPH)
5542        INPUT, ATR(IPH)
5562        GØ TØ 135
5582   140  CØNTINUE
5602        DØ 143 I=1,NPHP
5622        DØ 143 J=1,3
5642        IF(IPHASE(I,J))143,143,141
5662   141  DØ 142 K=1,NPHP
5682        IF(IPHASE(K,4)-IPHASE(I,J))142,143,142
5702   142  CØNTINUE
5722        GØ TØ 160
5742   143  CØNTINUE
5762   150  RETURN
5782   160  PRINT 703
5802   170  NPHP=0
5822        GØ TØ 150
5842   700  FØRMAT(I3," IS AN INVALID PHASE")
5862   701  FØRMAT(" FØLLØWING PHASES FØR",I3,",ARE",3I3/" PLEASE
5882       & CØRRECT (XX,XX,XX)")
5902   702  FØRMAT(" PHASE",I3," ATTRITIØN RATE ØF",F8.4/
5922       &" IS INVALID RE-ENTER THE CØRRECT VALUE (.XXX)")
5942   703  FØRMAT(" ALL PHASES DELETED")
5962        END
```

TABLE 11 (Cont)

e. Subroutine NOYES

```
5982        SUBRØUTINE NØYES
6002        CØMMØN SWITCH(11),NAME(25,3),SPACE(25,50)
6022        CØMMØN ICØMMA,IBLANK,NØ,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
6042       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSØUT(25),SØ(25)
6062        CØMMØN NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSØSW
6082       &,IDATA(5),IC
6102     10 I=1
6122        INPUT 700,NY
6142        IF(NØ-NY)30,20,30
6162     20 NY=-1*I
6182        RETURN
6202     30 I=-1
6222        IF(NYES-NY)40,20,40
6242     40 IER=7
6262        CALL ERRØR
6282        GØ TØ 10
6302    700 FØRMAT(A1)
6322        END
```

TABLE 11 (Cont)

f. Subroutine ERROR

```
6342        SUBRØUTINE ERRØR
6362        CØMMØN SWITCH(11),NAME(25,3),SPACE(25,50)
6382        CØMMØN ICØMMA,IBLANK,NØ,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
6402       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSØUT(25),SØ(25)
6422        CØMMØN NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSØSW
6442       &,IDATA(5),IC
6462        GØTØ (1,2,3,4,5,6,7,8,9,10),IER
6482      1 PRINT 701
6502        GØ TØ 100
6522      2 PRINT 702
6542        GØ TØ 100
6562      3 PRINT 703
6582        GØ TØ 100
6602      4 PRINT 704
6622        GØ TØ 100
6642      5 PRINT 705
6662        GØ TØ 100
6682      6 PRINT 706
6702        GØ TØ 100
6722      7 PRINT 707
6742        GØ TØ 100
6762      8 PRINT 708
6782        GØ TØ 100
6802      9 PRINT 709,IID,FID,(IDATA(J),J=1,IPH)
6822        PRINT 729
6842        GØ TØ 100
6862     10 PRINT 710
6882    100 IER=0
6902        RETURN
6922    701 FØRMAT(" PHASE IN PIPELINE")
6942    702 FØRMAT(" PHASE NØT IN PIPELINE")
6962    703 FØRMAT(" PIPELINE LØGIC ERRØR - ALL PHASES DELETED")
6982    704 FØRMAT(" NØ PHASES IN PIPELINE")
7002    705 FØRMAT(" MAXIMUM PHASES IN PIPELINE")
7022    706 FØRMAT(37H MAX. FØR FIELD IS 3 - FIELD SET TØ 0)
7042    707 FØRMAT(" INVALID REPLY - REPEAT")
7062    708 FØRMAT(23H CØMPUTER ERRØR, RE-RUN)
7082    709 FØRMAT(27H RESIDUAL ØUTPUT FRØM PHASE,I3,3H IS,F6.0,9H STUDEN
7102       &TS/34H DIVIDE AMØNG THE FØLLØWING PHASES,I3,2(A1,I2))
7122    710 FØRMAT(" INSUFFICIENT DATA TØ CØMPUTE STUDENT STATISTICS"/
7142       &" RE-ENTER STUDENT ASSIGNMENTS ØR RERUN")
7162    729 FØRMAT(" ENTER APRØPIATE MIX (XXXXX.,XXXXX.,XXXXX.)"/" ")
7182        END
```

TABLE 11 (Cont)

g. Subroutine PIPENT

```
7202       SUBRØUTINE PIPENT
7222       CØMMØN SWITCH(11),NAME(25,3),SPACE(25,50)
7242       CØMMØN ICØMMA,IBLANK,NØ,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
7262      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSØUT(25),SØ(25)
7282       CØMMØN NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSØSW
7302      &,IDATA(5),IC
7322     5 DØ 10 I=1,NPH
7342    10 SØ(I)=0.0
7362       PRINT 702,PNAME
7382    20 INPUT,IPH,SØUT
7402       IF(IPH)60,90,30
7422    30 IF(IPH-NPH)40,40,60
7442    40 DØ 50 I=1,NPHP
7462       IF(IPHASE(I,4)-IPH)50,70,50
7482    50 CØNTINUE
7502    60 IER=2
7522       CALL ERRØR
7542       GØ TØ 20
7562    70 IF(SØUT)60,80,80
7582    80 SØ(IPH)=-SØUT
7602       PRINT 703
7622       GØ TØ 20
7642    90 CALL ØUTPUT
7662       IF(IER)120,120,100
7682   100 CALL ERRØR
7702       GØ TØ 5
7722   120 CALL SMØØTH
7742       RETURN
7762   700 FØRMAT(5X,5E13.6)
7782 702 FØRMAT(//" FØR PIPELINE ",3A4/" ENTER PHASE NUMBER AND ST
7802      &UDENT ØUTPUT (XX,XXXX.)"/" PHASE 0,0 IMPLIES NØ FURTHER A
7822      &SSIGNMENTS")
7842   703 FØRMAT(5H NEXT)
7862       END
```

TABLE 11 (Cont)

h. Subroutine PIPINP

```
7882       SUBR0UTINE PIPINP
7902       C0MM0N SWITCH(11),NAME(25,3),SPACE(25,50)
7922       C0MM0N IC0MMA,IBLANK,N0,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
7942      &AFD,KILL,IID,FID,KILLS(25),SI(25),TS0UT(25),S0(25)
7962       C0MM0N NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LS0SW
7982      &,IDATA(5),IC
8002       FILENAME PIPE
8022       PIPE="PIPE"
8042       IF(NPSW)65,10,10
8062    10 READ(PIPE,700)NPHP,PNAME
8082       IF(NPHP)60,50,30
8102    30 D0 40 I=1,NPHP
8122       READ(PIPE,701)IL,(IPHASE(I,J),J=1,4),AT
8142       IPH=IPHASE(I,4)
8162    40 ATR(IPH)=AT
8182       G0 T0 70
8202    50 PRINT 702,PNAME
8222       NPHP=0
8242       CALL N0YES
8262       IF(NY)60,60,120
8282    60 NPSW=NPHP
8302    65 RETURN
8322    70 IF(KILL)90,90,80
8342    80 D0 85 I=1,KILL
8362       IID=-1
8382       IPH=KILLS(I)
8402    85 CALL DPIPE
8422       IID=0
8442       IF(NPHP)50,50,90
8462    90 IF(IPRT)110,110,100
8482   100 CALL PIPRT
8502   110 IF(LEVLSR-2)130,130,120
8522   120 CALL MPIPE
8542   130 CALL PIPER
8562       IF(NPHP)50,50,140
8582   140 IF(NPHP-NPH)150,150,50
8602   150 CALL L0ADS0
8622       IF(NPHP)50,50,60
8642   700 F0RMAT(5X,I3,3A4)
8662   701 F0RMAT(V)
8682   702 F0RMAT(31H N0 PHASES EXIST F0R PIPELINE ~,3A4/21H ENTER NEW
8702      & DATA (Y,N))
8722       END
```

AD-A043 864 OPERATIONS RESEARCH INC SILVER SPRING MD F/G 5/1
DEVELOPMENT OF A PRELIMINARY AUTOMATED TOTAL SYSTEMS MODEL FOR --ETC(U)
FEB 70 T N KYLE, R D HEILBRON, J D AVILA N00025-67-C-0031
UNCLASSIFIED ORI-TR-583-VOL-4 NL

2 OF 3
AD A043864

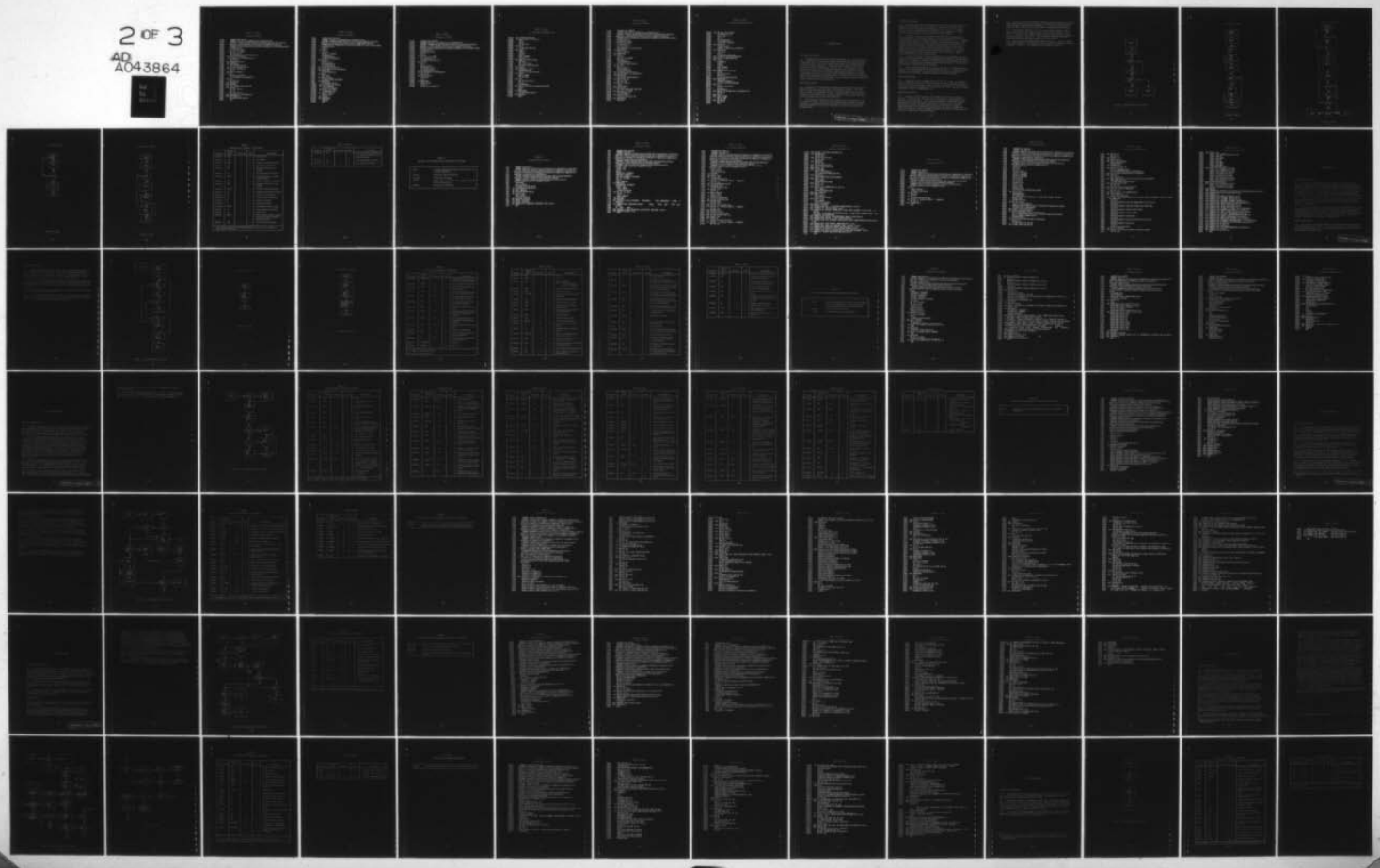


TABLE 11 (Cont)

i. Subroutine DPIPE

```
8742        SUBRØUTINE DPIPE
8762        CØMMØN SWITCH(11),NAME(25,3),SPACE(25,50)
8782        CØMMØN ICØMMA,IBLANK,NØ,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
8802       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSØUT(25),SØ(25)
8822        CØMMØN NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSØSW
8842       &,IDATA(5),IC
8862        IF(IPH)70,70,5
8882      5 DØ 30 I=1,NPHP
8902        DØ 30 J=1,4
8922        IF(IPHASE(I,J)-IPH)30,20,10
8942     10 IF(IID)15,30,30
8962     15 IPHASE(I,J)=IPHASE(I,J)-1
8982        GØ TØ 30
9002     20 IPHASE(I,J)=0
9022     30 CØNTINUE
9042        IF(IID)40,70,70
9062     40 IF(IPH-25)50,70,70
9082     50 DØ 60 I=IPH,24
9102        K=I+1
9122     60 ATR(I)=ATR(K)
9142     70 L=0
9162        DØ 90 I=1,NPHP
9182        IF(IPHASE(I,4))80,80,90
9202     80 L=I
9222        GØ TØ 110
9242     90 CØNTINUE
9262    100 RETURN
9282    110 IF(L-NPHP)120,140,140
9302    120 M=NPHP-1
9322        DØ 130 I=L,M
9342        K=I+1
9362        DØ 130 J=1,4
9382    130 IPHASE(I,J)=IPHASE(K,J)
9402    140 NPHP=NPHP-1
9422        IF(NPHP)100,100,70
9442        END
```

TABLE 11 (Cont)

j. Subroutine OUTPUT

```
9462        SUBROUTINE OUTPUT
9482        COMMON SWITCH(11),NAME(25,3),SPACE(25,50)
9502        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
9522       &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
9542        COMMON NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSOSW
9562       &,IDATA(5),IC
9582        IER=0
9602    10  ICK=0
9622        IALL=0
9642        DO 60 L=1,NPHP
9662        M=NPHP+1-L
9682        I=IPHASE(M,4)
9702        IF(SO(I))50,20,20
9722    20  TA=0.0
9742        DO 40 J=1,3
9762        K=IPHASE(M,J)
9782        IF(K)40,40,30
9802    30  IF(SO(K))35,60,60
9822    35  TA=TA+SO(K)/(1.0-ATR(K))
9842    40  CONTINUE
9862        IF(TA)45,60,60
9882    45  ICK=1
9902        SO(I)=TA
9922    50  IALL=IALL+1
9942        IF(IALL-NPHP)60,80,80
9962    60  CONTINUE
9982        IF(ICK)70,70,10
10002   70  IF(LSOSW)75,75,90
10022   75  CALL OUTFOR
10042       IF(NY)90,90,10
10062   80  DO 85 L=1,NPHP
10082       I=IPHASE(L,4)
10102   85  SO(I)=-SO(I)
10122       GO TO 100
10142   90  IER=10
10162   100 CONTINUE
10182       RETURN
10202       END
```

TABLE 11 (Cont)

k. Subroutine OUTFOR

```
10222       SUBRØUTINE ØUTFØR
10242       CØMMØN SWITCH(11),NAME(25,3),SPACE(25,50)
10262       CØMMØN ICØMMA,IBLANK,NØ,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
10282      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSØUT(25),SØ(25)
10302       CØMMØN NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSØSW
10322      &,IDATA(5),IC
10342       DIMENSIØN T(3)
10362       NY=0
10382    10 ICK=0
10402       DØ 170 II=1,NPHP
10422       I=IPHASE(II,4)
10442       IF(SØ(I))20,170,170
10462    20 TA=0.0
10482       IL=1
10502       DØ 60 J=1,3
10522       K=IPHASE(II,J)
10542       IF(K)60,60,30
10562    30 IF(SØ(K))40,50,50
10582    40 TA=TA+SØ(K)/(1.0-ATR(K))
10602       GØ TØ 60
10622    50 IDATA(IL)=K
10642       IL=IL+2
10662       L=K
10682    60 CØNTINUE
10702       A=SØ(I)-TA
10722       IL=IL-2
10742       IF(IL-1)170,80,110
```

TABLE 11 (Cont)

k. Subroutine OUTFOR (Cont)

```
10762    80 IF(A)90,180,180
10782    90 S0(L)=A*(1.0-ATR(L))
10802       NY=1
10822       ICK=1
10842       G0 T0 170
10862   110 A=-A
10882       IF(A)180,180,120
10902   120 IID=I
10922       IER=9
10942       IPH=IL
10962       FID=A
10982       CALL ERR0R
11002       N=IL/2 + 1
11022   125 INPUT,(T(J),J=1,N)
11042       T0T=0.0
11062       D0 130 J=1,N
11082       IF(T(J))140,140,130
11102   127 IER=2
11122   130 T0T=T0T + T(J)
11142       R=ABS(T0T-A)
11162       IF(R-1.5)150,150,140
11182   140 IER=7
11202       CALL ERR0R
11222       G0 T0 125
11242   150 J=0
11262       D0 160 L=1,IL,2
11282       J=J+1
11302       K=IDATA(L)
11322   160 S0(K)=-T(J)*(1.0-ATR(K))*A/T0T
11342       ICK=1
11362       NY=1
11382   170 C0NTINUE
11402       IF(ICK)180,180,10
11422   180 RETURN
11442       END
```

TABLE 11 (Cont)

1. Subroutine SMOOTH

```
11462       SUBROUTINE SMOOTH
11482       COMMON SWITCH(11),NAME(25,3),SPACE(25,50)
11502       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
11522      &AFD,KILL,IID,FID,KILLS(25),SI(25),TSOUT(25),SO(25)
11542       COMMON NPHP,IPHASE(25,4),ATR(25),PNAME(3),IPRT,NPSW,LSOSW
11562      &,IDATA(5),IC
11582       DIMENSION T(3)
11602     5 DO 20 L=1,NPHP
11622       I=IPHASE(L,4)
11642       DO 10 J=1,3
11662       IF(IPHASE(L,J))10,10,20
11682    10 CONTINUE
11702       SO(I)=-SO(I)
11722    20 CONTINUE
11742    30 ICK=0
11762       IALL=0
11782       DO 110 L=1,NPHP
11802       M=NPHP+1-L
11822       I=IPHASE(M,4)
11842       IF(SO(I))100,100,40
11862    40 TA=.01
11882       DO 70 J=1,3
11902       K=IPHASE(M,J)
11922       IF(K)70,70,50
11942    50 IF(SO(K))60,70,110
11962    60 TA=TA+SO(K)/(1.0-ATR(K))
11982    70 CONTINUE
12002       IF(SO(I)+TA)90,80,80
12022    80 SO(I)=TA-.01
12042       GO TO 100
12062    90 SO(I)=-SO(I)
12082       ICK=1
12102   100 IALL=IALL+1
12122   110 CONTINUE
12142       IF(IALL-NPHP)30,120,120
12162   120 DO 130 L=1,NPHP
12182       I=IPHASE(L,4)
12202   130 SO(I)=-SO(I)
12222       IF(ICK)140,140,150
12242   140 CONTINUE
12262       RETURN
```

TABLE 11 (Cont)

1. Subroutine SMOOTH (Cont)

```
12282  150 DØ 300 II=1,NPHP
12302      I=IPHASE(II,4)
12322      TA= -.01
12342      IL=-1
12362      DØ 170 J=1,3
12382      K=IPHASE(II,J)
12402      IF(K)170,170,160
12422  160 IL=IL+2
12442      IDATA(IL)=K
12462      TA=TA + SØ(K)/(1.0-ATR(K))
12482  170 CØNTINUE
12502      T(1)=1.0
12522      R=1.0
12542      IF(IL)300,300,180
12562  180 IF(SØ(L)-TA)190,300,300
12582  190 IF(IL-1)300,250,195
12602  195 IID=L
12622      FID=SØ(L)
12642      IER=9
12662      IPH=IL
12682      CALL ERRØR
12702      N=IL/2 + 1
12722  200 INPUT,(T(I),I=1,N)
12742      R=0.0
12762      DØ 240 I=1,N
12782      R=R+T(I)
12802      IF(T(I))280,240,240
12822  240 CØNTINUE
12842      TA=ABS(R-SØ(L))
12862      IF(TA-1.5)250,250,280
12882  250 I=0
12902      DØ 260 J=1,IC,2
12922      I=I+1
12942      K=IDATA(J)
12962  260 SØ(K)=T(I)*SØ(L)*(1.0-ATR(K))/R
12982      GØ TØ 5
13002  280 IER=7
13022  290 CALL ERRØR
13042      GØ TØ 200
13062  300 CØNTINUE
13082      GØ TØ 5
13102      END
```

V. PROGRAM LSR3

PROGRAM DESCRIPTION

5.1 PROGRAM LSR3 develops LSR output statements, i.e., resource requirements associated with the student outputs for each training phase developed in PROGRAM LSR2. Upon entry, control is transferred to Subroutine LSTLSR to print the LSR output statements. When control is returned, the LSR level of complexity, LEVLSR, is tested for a value of 2 or 4. Should either value be present, control is passed to Subroutine MODLSR which constrains the LSR output due to limited training resources. When control returns or if the level of complexity is not 2 or 4, the user is asked whether he desires to generate another LSR output based on a different PTR. If another LSR output is requested, control is transferred to PROGRAM LSR1; otherwise control is passed to PROGRAM LSR4 to compute the runway requirements.

SUBROUTINE LSTLSR

5.2 Subroutine LSTLSR develops the necessary program linkage to print LSR output statements, i.e., either detail or summary and develop a data file, LSROUT, of the LSR information for use in subsequent programs. Upon entry, data file LSROUT is opened and rewound. Next the number of training phases is written on data file LSROUT. Subroutine GENLSR is then called sequentially for each training phase to develop the LSR data.

5.3 After Subroutine GENLSR has been called for each training phase, data file LSROUT is closed. The user is then asked whether a detailed LSR output is desired. If the detailed LSR data are requested, control is again passed to Subroutine GENLSR for each phase of training. Control is then returned to PROGRAM LSR3.

SUBROUTINE MODLSR

5.4 Subroutine MODLSR develops constrained LSR data resulting from limited training resources which are user-supplied. Upon entry, the user is asked whether he wishes to constrain the LSR output. With a Yes, "Y," response, the user inputs the number of the training phase to be constrained. A check is made to determine the validity of the phase number entered.

5.5 Subroutine GENLSR is called to develop a detailed LSR output for the training phase the user wishes to constrain. The user must then enter the type of constraint he wishes to impose on the selected training phase, i.e., constrain aircraft, flight instructors, enlisted personnel, or academic instructors. Subroutine MODLSR validates the user response and requests the user to enter the amount of the limiting resource available. For each resource constrained, a computation is made to determine the total number of students, i.e., student output, which can be annually trained with the limited resource. The results of this computation are then displayed for the user. The user may impose additional constraints on the training phase.

5.6 When all constraints have been imposed on the training phase, the user is asked whether he wishes a detailed LSR output for the phase he has constrained. With a Yes, "Y," response, Subroutine GENLSR is called to develop the detailed LSR output which will not exceed the most limiting training resource. The user may then constrain additional phases of training.

5.7 When no further constraints are desired, the user may constrain all training phases by adjusting the PTR in each training pipeline, i.e., for each student source. If the user wishes to exercise this option, control is transferred to PROGRAM LSR2; otherwise, control is returned to the calling routine.

SUBROUTINE NOYES

5.8 Subroutine NOYES is designed to read a Yes, "Y," or No, "N," response from the time-sharing terminal. Switch NY, which is carried through common, is set to -1 with a No reply and to 1 for a Yes reply. Whenever a Yes or No response is required, control is transferred to Subroutine NOYES, which immediately returns control with Switch NY appropriately set.

SUBROUTINE GENLSR

5.9 Subroutine GENLSR generates LSR summary statements for a phase of training. Upon entry, the LSR output data are initialized. The student load for the phase is then computed. A test is made to determine whether the training phase contains flight instruction. If flight training is included in the phase, the number of aircraft, landing support officers, gallons of fuel consumed, flight instructors, flight instructors under training, and enlisted support personnel are computed. Enlisted support personnel requirements are then increased to include administrative enlisted personnel.

5.10 A test is next made to determine whether academic instruction is included in the training phase. Required academic instructors and academic instructors under training are computed when appropriate. The number of administrative officers is then computed from the total number of phase personnel required.

5.11 Depending on Switch NY, which is carried through common, either a summary or a detailed LSR output is printed for the training phases. With NY = -1, a one-line LSR summary is printed and the total LSR output is written on data file LSROUT. When NY = 1, only a detailed LSR output is printed. Subroutine GENLSR then returns control to the calling routine.

5.12 A flow chart of PROGRAM LSR3 is shown in Figure 5. Table 12 contains the variable dictionary of PROGRAM LSR3; the program and subroutine dictionary is provided in Table 13. The program listing is shown in Table 14.

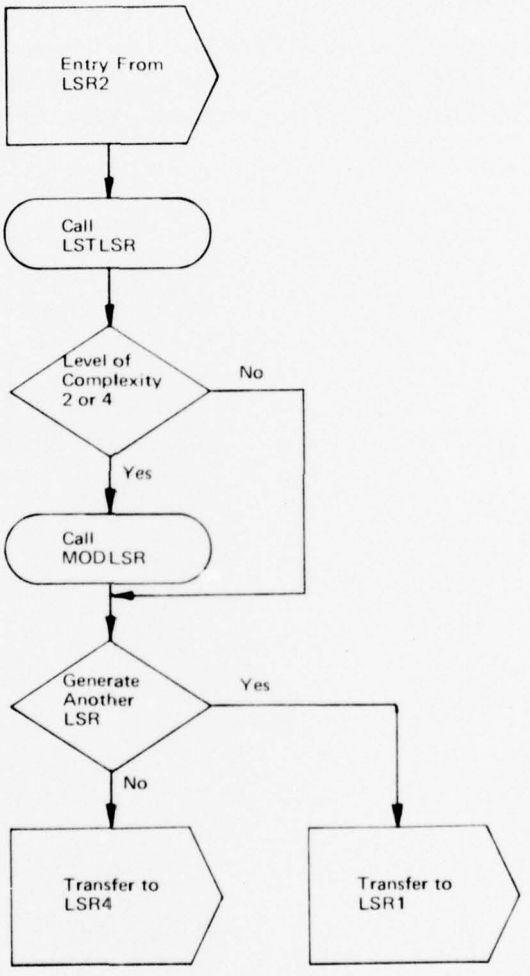


FIGURE 5. PROGRAM LSR3 FLOW CHART

a. Subroutine LSTLSR

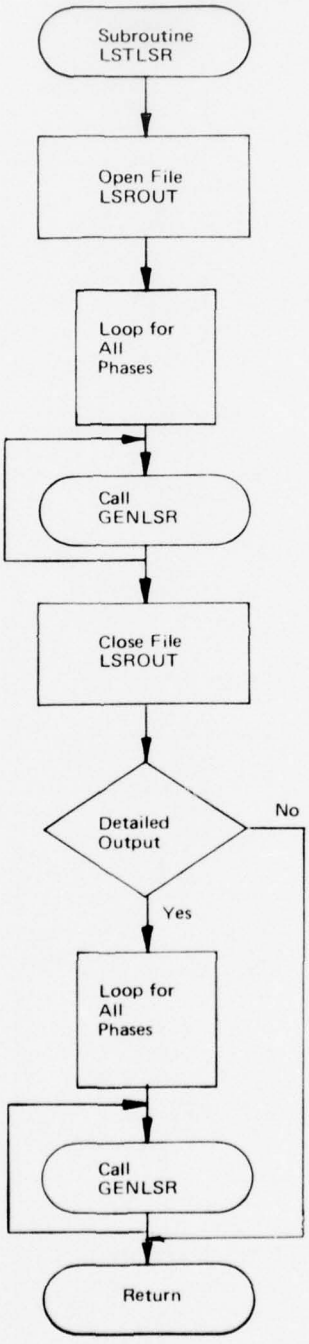


FIGURE 5 (Cont)

b. Subroutine MODLSR

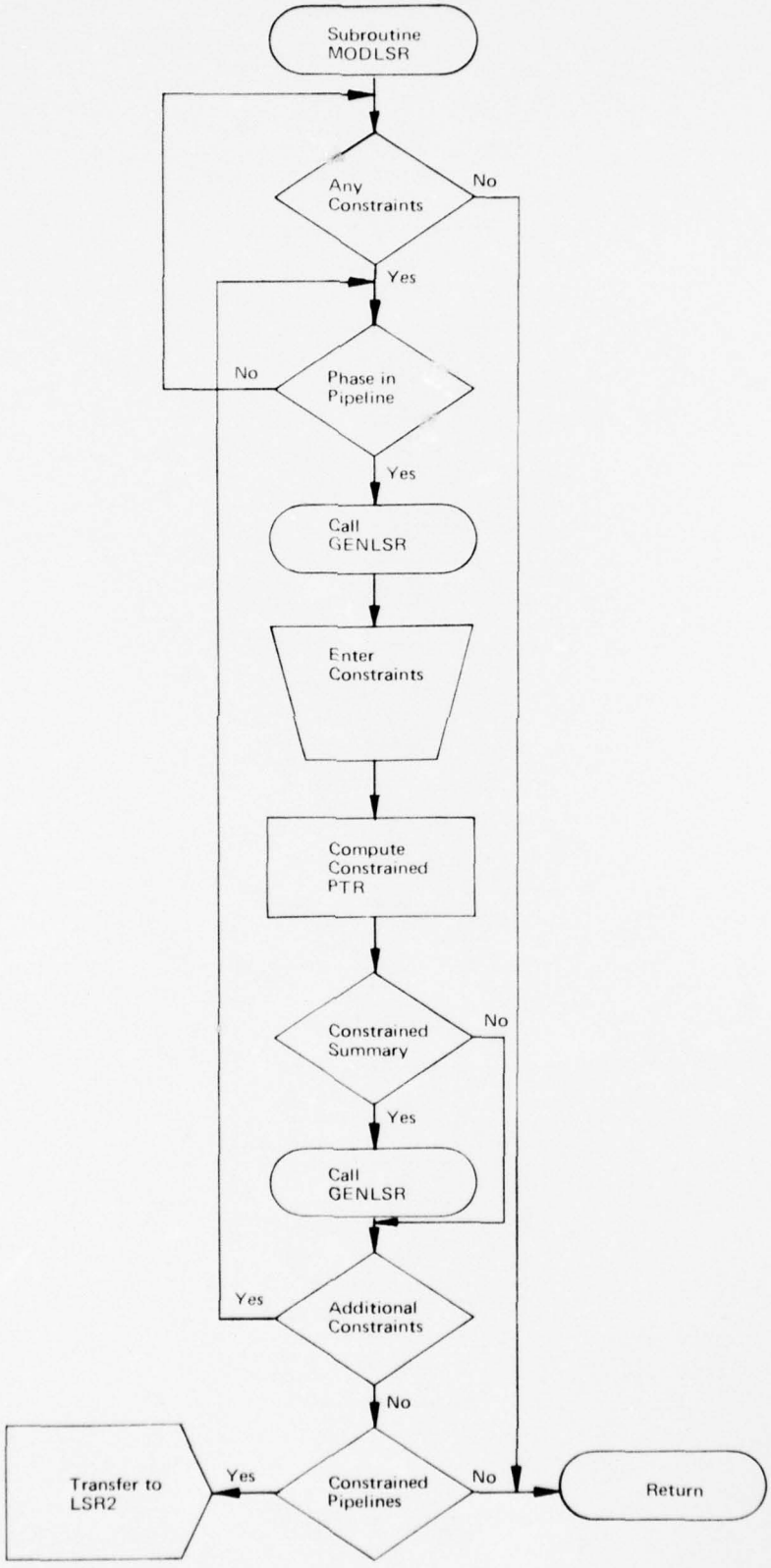


FIGURE 5 (Cont)

c. Subroutine NOYES

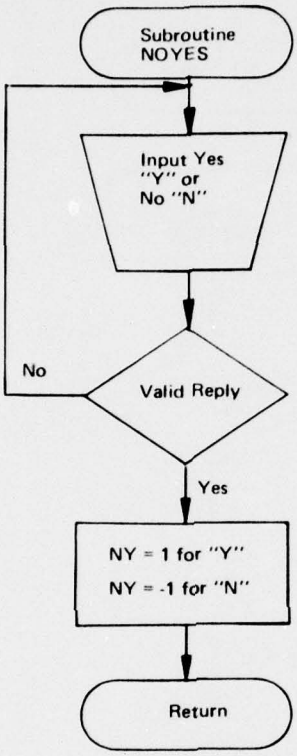


FIGURE 5 (Cont)

d. Subroutine GENSLR

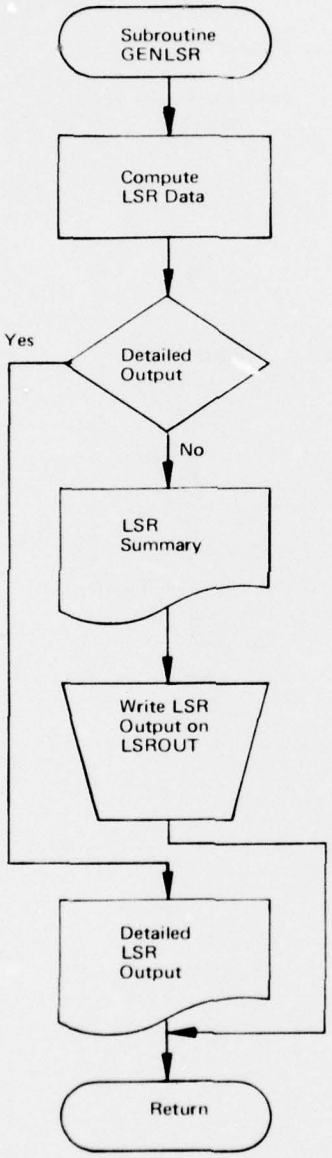


FIGURE 5 (Cont)

TABLE 12

PROGRAM LSR3 VARIABLE DICTIONARY*

Location	Variable Name	Dimension	Type	Description
Common	IACT	3	A	Aircraft types
Common	IAFT	3	A	Fuel types
Common	IAIN	3	A	Academic instruction types
Common	BF	3	F	Bulk fuel requirements
Common	FIT	3	F	Flight instructors under training
Common	FI	3	F	Required flight instructors
Common	FLSO	3	F	Required number of landing support officers
Common	EM	3	F	Enlisted support requirements
Common	AIT	3	F	Academic instructors under training
Common	ACNO	3	F	Number of aircraft required
Common	AI	3	F	Academic instructors required
MODLSR	IF	1	I	Field to be constrained
MODLSR	IE	1	I	Element to be constrained
MODLSR	D	1	F	Resource available
MODLSR	V	1	F	Resource required
MODLSR	SOUT	1	F	Constrained student output
GENLSR	U	3	F	Annual aircraft utilization
GENLSR	SL	1	F	Student load
GENLSR	FACT	1	F	Ratio of total enlisted support required to enlisted maintenance personnel
GENLSR	EMT	1	F	Total enlisted personnel

* See variable dictionary for PROGRAM LSRM in Table 3 for description of other common variables.

TABLE 12 (Cont)

Location	Variable Name	Dimension	Type	Description
GENLSR	TSP	1	F	Ratio of number of administrative officers required to total phase personnel
GENLSR	AM	1	F	Total administrative officers
GENLSR	IC	1	F	Line number for LSROUT

TABLE 13

PROGRAM LSR3 PROGRAM AND SUBROUTINE DICTIONARY

LSR3	Develops appropriate program linkage to list LSR summary statements
LSTLSR	Prints LSR summary headings
MODLSR	Constrains LSR output
NOYES	Reads a Yes, "Y," or No, "N," response from the time-sharing terminal
GENLSR	Develops LSR output data

TABLE 14

PROGRAM LSR3 LISTING

```
103       COMMON SWITCH(11)
123       COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
143      &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
163      &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
183      &ASH(25,3),AIH(25,3),AITR(25,3)
203       COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
223      &AFD,KILL,IID,FID,KILLS(25),SINP(25),SO(25)
243       COMMON IACT(3),IAFT(3),IAIN(3),BF(3),FIT(3),FI(3),
263      &FLSO(3),EM(3),AIT(3),ACNO(3),AI(3)
283       IID=1000
303       CALL LSTLSR
323       IF(LEVLSR-2)30,20,10
343    10 IF(LEVLSR-4)30,20,30
363    20 CALL MODLSR
383    30 IF(LEVLSR)50,50,40
403    40 PRINT 700
423       CALL NOYES
443       IF(NY)60,60,70
463    50 LEVLSR=-LEVLSR
483    60 CHAIN "XLSR4*"
503    70 CHAIN "XLSR1*"
523   700 FORMAT(27H GENERATE ANOTHER LSR (Y,N))
543       END
```

TABLE 14 (Cont)

a. Subroutine LSTLSR

```
      SUBROUTINE LSTLSR
      COMMON SWITCH(11)
      COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
     &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
     &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
     &ASH(25,3),AIH(25,3),AITR(25,3)
      COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
     &AFD,KILL,IID,FID,KILLS(25),SINP(25),SO(25)
      COMMON IACT(3),IAFT(3),IAIN(3),BF(3),FIT(3),FI(3),
     &FLSO(3),EM(3),AIT(3),ACNO(3),AI(3)
      PRINT 700
      PRINT 701
      NY=0
      OPENFILE "LSROUT"
      REWIND "LSROUT"
      WRITE("LSROUT",703)NPH
      DO 10 I=1,NPH
      IPH=I
      CALL GENLSR
   10 CONTINUE
      CLOSEFILE "LSROUT"
      PRINT 702
       CALL NOYES
       IF(NY)40,40,20
   20 DO 30 I=1,NPH
       IPH=I
       CALL GENLSR
   30 CONTINUE
   40 RETURN
  700 FORMAT(//19X,"STUDENT   AIRCRAFT     FUEL CONSUMED   TOTAL  T
     &OTAL")
  701 FORMAT(69H TRAINING PHASE      LOAD    TYPE   NO.   TYPE  GAL
     &LONS
     &   OFF     ENL)
  702 FORMAT(//34H DETAILED LSR OUTPUT DESIRED (Y,N))
  703 FORMAT(5H1000 ,I3)
      END
```

TABLE 14 (Cont)

b. Subroutine MODLSR

```
1303        SUBROUTINE MODLSR
1323        COMMON SWITCH(11)
1343        COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
1363       &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
1383       &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
1403       &ASH(25,3),AIH(25,3),AITR(25,3)
1423        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
1443       &AFD,KILL,IID,FID,KILLS(25),SINP(25),SO(25)
1463        COMMON IACT(3),IAFT(3),IAIN(3),BF(3),FIT(3),FI(3),
1483       &FLSO(3),EM(3),AIT(3),ACNO(3),AI(3)
1503      5 PRINT 700
1523        CALL NOYES
1543        IF(NY)10,10,20
1563     10 RETURN
1583     20 PRINT 701
1603     40 INPUT,IPH
1623        IF(IPH)50,260,60
1643     50 PRINT," INVALID REPLY - REPEAT"
1663        GO TO 40
1683     60 IF(IPH-NPH)65,65,50
1703     65 SOUT=SO(IPH)
1723        IF(SOUT)66,66,67
1743     66 PRINT 714
1763        GO TO 5
1783     67 CALL GENLSR
1803        PRINT 703
1823     70 INPUT,IF,IE
1843        IF(IF)110,250,80
1863     80 IF(IF-3)90,90,100
1883     90 N=NAC(IPH)
1903        GO TO 130
1923    100 IF(IF-4)110,120,110
1943    110 PRINT," INVALID REPLY - REPEAT"
1963        GO TO 70
1983    120 N=NAD(IPH)
2003    130 IF(N)5,5,140
2023    140 IF(IE)110,110,145
2043    145 IF(IE-N)150,150,110
2063    150 PRINT 705
2083    155 INPUT,D
2103        IF(D-0.1)157,157,160
2123    157 PRINT," INVALID REPLY - REPEAT"
2143        GO TO 155
```

TABLE 14 (Cont)

b. Subroutine MODLSR (Cont)

```
2163   160 GØ TØ (170,180,190,200),IF
2183   170 V=ACNØ(IE)
2203       GØ TØ 210
2223   180 V=FIT(IE)+FI(IE)
2243       GØ TØ 210
2263   190 V=EM(IE)
2283       GØ TØ 210
2303   200 V=AIT(IE)+AI(IE)
2323   210 IF(D-V)230,220,220
2343   220 PRINT 707
2363   230 S=D/V*SØUT
2383       PRINT 708,SØUT,S
2403       IF(S-SØ(IPH))240,240,250
2423   240 SØ(IPH)=S
2443       SINP(IPH)=SINP(IPH)*S/SØUT
2463   250 PRINT 709
2483       CALL NØYES
2503       IF(NY)255,255,252
2523   252 PRINT 713
2543       GØ TØ 70
2563   255 PRINT 710,(NAME(IPH,J),J=1,3)
2583       CALL NØYES
2603       IF(NY)260,260,258
2623   258 CALL GENLSR
2643   260 PRINT 711
2663       CALL NØYES
2683       IF(NY)270,270,20
2703   270 PRINT 712
2723       CALL NØYES
2743       IF(NY)10,10,280
2763   280 CHAIN "XLSR2*"
2783   700 FØRMAT(33H ANY LSR ØUTPUT CØNSTRAINTS (Y,N))
2803   701 FØRMAT(17H WHICH PHASE (XX))
2823   703 FØRMAT(41H SELECT APRØPIATE FIELD AND ELEMENT (X,X)/12H  1 A
2843      &IRCRA
2863      &FT/22H  2 FLIGHT INSTRUCTØRS/20H  3 ENLISTED SUPPØRT/24H  4 A
2883      &CADEMIC INSTRUCTØRS)
2903   705 FØRMAT(36H ENTER CØNSTRAINING VALUE (XXXX.XXX))
2923   707 FØRMAT(26H VALUE IS NØT CØNSTRAINING)
2943   708 FØRMAT(19H ØLD STUDENT ØUTPUT,F6.0/19H CØNSTRAINED ØUTPUT,F6.
2963      &0)
2983   709 FØRMAT(29H ADDITIØNAL CØNSTRAINTS (Y,N))
3003   710 FØRMAT(21H NEW LSR SUMMARY FØR ,3A4,6H (Y,N))
3023   711 FØRMAT(32H ANØTHER PHASE CØNSTRAINED (Y,N))
3043   712 FØRMAT(" REVISE LSR TØ INCLUDE CØNSTRAINTS (Y,N)")
3063   713 FØRMAT(42H SELECT APPRØPIATE FIELD AND ELEMENT (X,X))
3083   714 FØRMAT(" PHASE CØNTAINS NØ ACTIVITY")
3103       END
```

TABLE 14 (Cont)

c. Subroutine NOYES

```
3123        SUBROUTINE NOYES
3143        COMMON SWITCH(11)
3163        COMMON NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
3183       &WK(25),TOD(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
3203       &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FSO(25,3),AMO(25,3),
3223       &ASH(25,3),AIH(25,3),AITR(25,3)
3243        COMMON ICOMMA,IBLANK,NO,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
3263       &AFD,KILL,IID,FID,KILLS(25),SINP(25),SO(25)
3283        COMMON IACT(3),IAFT(3),IAIN(3),BF(3),FIT(3),FI(3),
3303       &FLSO(3),EM(3),AIT(3),ACNO(3),AI(3)
3323    10  I=1
3343        INPUT 700,NY
3363        IF(NO-NY)30,20,30
3383    20  NY=-1*I
3403        RETURN
3423    30  I=-1
3443        IF(NYES-NY)40,20,40
3463    40  PRINT," INVALID REPLY - REPEAT"
3483        GO TO 10
3503   700  FORMAT(A1)
3523        END
```

TABLE 14 (Cont)

d. Subroutine GENLSR

```
3543        SUBR0UTINE GENLSR
3563        C0MM0N SWITCH(11)
3583        C0MM0N NAME(25,3),NPLA(25,3),NFUEL(25,3),NACD(25,3),ATP(25),
3603       &WK(25),T0D(25),NAC(25),NAD(25),WX(25,3),GAS(25,3),AU(25,3),
3623       &FU(25,3),SFH(25,3),FIH(25,3),FTR(25,3),FS0(25,3),AM0(25,3),
3643       &ASH(25,3),AIH(25,3),AITR(25,3)
3663        C0MM0N IC0MMA,IBLANK,N0,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
3683       &AFD,KILL,IID,FID,KILLS(25),SINP(25),S0(25)
3703        C0MM0N IACT(3),IAFT(3),IAIN(3),BF(3),FIT(3),FI(3),
3723       &FLS0(3),EM(3),AIT(3),ACN0(3),AI(3)
3743        DIMENSI0N U(3)
3763        EMT=0.0
3783        IC=IID
3803        D0 10 I=1,3
3823        IACT(I)=IBLANK
3843        IAFT(I)=IBLANK
3863        IAIN(I)=IBLANK
3883        BF(I)=0.0
3903        FIT(I)=0.0
3923        FI(I)=0.0
3943        FLS0(I)=0.0
3963        EM(I)=0.0
3983        AIT(I)=0.0
4003        ACN0(I)=0.0
4023        U(I)=AU(IPH,I)*WX(IPH,I)*AFD
4043     10 AI(I)=0.0
4063        S0UT=S0(IPH)
4083        SI=SINP(IPH)
4103        SL=(SI*ATP(IPH)+S0UT*(1.0-ATP(IPH)))*WK(IPH)/WPY
4123        N=NAC(IPH)
4143        IF(N)95,95,20
4163     20 D0 30 I=1,N
4183        IACT(I)=NPLA(IPH,I)
4203        ACN0(I)=(S0UT*SFH(IPH,I))/(AU(IPH,I)*WX(IPH,I)*AFD)
4223        IF(FS0(IPH,I))28,28,24
4243     24 FLS0(I)=SL/FS0(IPH,I)
4263     28 IAFT(I)=NFUEL(IPH,I)
4283        BF(I)=S0UT*GAS(IPH,I)*SFH(IPH,I)
4303        FI(I)=(S0UT*FIH(IPH,I))/(FU(IPH,I)*WX(IPH,I)*AFD)
4323        EM(I)=ACN0(I)*AM0(IPH,I)
4343        EMT=EMT+EM(I)
4363     30 FIT(I)=FI(I)*FTR(IPH,I)/T0D(IPH)
4383        FACT=1.2
4403        IF(EMT-200.)70,50,40
4423     40 IF(EMT-400.)50,60,60
```

TABLE 14 (Cont)

d. Subroutine GENLSR (Cont)

```
4443    50 FACT=1.15
4463       GØ TØ 70
4483    60 FACT=1.10
4503    70 DØ 80 I=1,N
4523    80 EM(I)=EM(I)*FACT
4543       EMT=FACT*EMT
4563    95 M=NAD(IPH)
4583       IF(M)120,120,100
4603   100 DØ 110 I=1,M
4623       IAIN(I)=NACD(IPH,I)
4643       AI(I)=SØUT*ASH(IPH,I)/AIH(IPH,I)
4663   110 AIT(I)=AI(I)*AITR(IPH,I)/TØD(IPH)
4683   120 TØFF=0.0
4703       DØ 140 I=1,3
4723   140 TØFF=TØFF+AI(I)+AIT(I)+FI(I)+FIT(I)+FLSØ(I)
4743       TSP=TØFF+EMT+SL
4763       IF(TSP-560.0)142,142,144
4783   142 AM=0.0303571*TSP
4803       GØ TØ 148
4823   144 IF(TSP-1260.0)146,146,147
4843   146 AM=7.4 + 0.0171428*TSP
4863       GØ TØ 148
4883   147 AM=17.8833 + 0.0088235*TSP
4903   148 TØFF=TØFF+AM
4923       IF(NY)150,150,180
4943   150 PRINT 700,(NAME(IPH,J),J=1,3),SL,IACT(1),ACNØ(1),IAFT(1),BF(1
4963      &),TØFF,EMT
4983       IC=IC+5
5003       WRITE("LSRØUT",719)IC,(NAME(IPH,J),J=1,3),N
5023       IC=IC+5
5043       WRITE("LSRØUT",720)IC,SI,SØUT,SL,TØFF,EMT
5063       IC=IC+5
5083       WRITE("LSRØUT",722)IC,IACT,IAFT
5103       IC=IC+5
5123       WRITE("LSRØUT",723)IC,ACNØ
5143       IC=IC+5
5163       WRITE("LSRØUT",723)IC,BF
5183       IC=IC+5
5203       WRITE("LSRØUT",723)IC,(ASH(IPH,J),J=1,3)
5223       IC=IC+5
5243       WRITE("LSRØUT",723)IC,U
5263       IID=IC
5283       IF(N-1)220,220,160
5303   160 DØ 170 I=2,N
5323   170 PRINT 701,IACT(I),ACNØ(I),IAFT(I),BF(I)
5343       GØ TØ 220
```

TABLE 14 (Cont)

d. Subroutine GENLSR (Cont)

```
5363   180 PRINT 702
5383       PRINT 703,(NAME(IPH,J),J=1,3)
5403       PRINT 704,SI
5423       PRINT 705,SØUT
5443       PRINT 706,SL
5463       PRINT 707,AM
5483       PRINT 708,TØFF
5503       PRINT 709,EMT
5523       IF(N)200,200,190
5543   190 PRINT 710,(IACT(I),I=1,N)
5563       PRINT 711,(ACNØ(I),I=1,N)
5583       PRINT 712,(IAFT(I),I=1,N)
5603       PRINT 713,(BF(I),I=1,N)
5623       PRINT 714,(FI(I),I=1,N)
5643       PRINT 715,(FIT(I),I=1,N)
5663       PRINT 721,(FLSØ(I),I=1,N)
5683       PRINT 716,(EM(I),I=1,N)
5703   200 IF(M)220,220,210
5723   210 PRINT 717,(IAIN(I),I=1,M)
5743       PRINT 718,(AI(I),I=1,M)
5763       PRINT 715,(AIT(I),I=1,M)
5783   220 RETURN
5803   700 FØRMAT(1X,3A4,F12.0,4X,A4,F6.0,3X,A4,E10.3,F6.0,F7.0)
5823   701 FØRMAT(29X,A4,F6.0,3X,A4,E10.3)
5843   702 FØRMAT(//)
5863   703 FØRMAT(15H NAME ØF PHASE ,3A4)
5883   704 FØRMAT(14H STUDENT INPUT,F6.0)
5903   705 FØRMAT(15H STUDENT ØUTPUT,F6.0)
5923   706 FØRMAT(21H AVERAGE STUDENT LØAD,F6.0)
5943   707 FØRMAT(24H ADMINISTRATIVE ØFFICERS,F6.0)
5963   708 FØRMAT(15H TØTAL ØFFICERS,F6.0)
5983   709 FØRMAT(15H TØTAL ENLISTED,F6.0)
6003   710 FØRMAT(15H AIRCRAFT TYPES,7X,3(1X,A4,4X))
6023   711 FØRMAT(16H NUMBER REQUIRED,F11.0,2F9.0)
6043   712 FØRMAT(11H FUEL TYPES,12X,A4,4X,A4,5X,A4)
6063   713 FØRMAT(17H GALLØNS CØNSUMED,3X,3E9.3)
6083   714 FØRMAT(19H FLIGHT INSTRUCTØRS,F8.0,2F9.0)
6103   715 FØRMAT(15H UNDER TRAINING,F12.0,2F9.0)
6123   716 FØRMAT(17H ENLISTED SUPPØRT,F10.0,2F9.0)
6143   717 FØRMAT(23H ACADEMIC INSTRUCTIØN  ,A4,2(5X,A4))
6163   718 FØRMAT(21H ACADEMIC INSTRUCTØRS,F6.0,2F9.0)
6183   719 FØRMAT(I4,1X,3A4,I3)
6203   720 FØRMAT(I4,1X,5E13.6)
6223   721 FØRMAT(17H LSØ REQUIREMENTS,F10.0,2F9.0)
6243   722 FØRMAT(I4,1X,6A4)
6263   723 FØRMAT(I4,1X,3E13.6)
6283       END
```

VI. PROGRAM LSR4

PROGRAM DESCRIPTION

6.1 The purpose of PROGRAM LSR4 is to develop the necessary program linkage for the computation of runway requirements, print the runway requirements, and generate the runway output file, RUNWAY. Upon entry, all variables are initialized for they partially overlay the common area which was utilized by the previous four programs. Subroutine INPRWY is then sequentially called to enter the runway data for each training phase containing flight instruction. For example, AOC school would be bypassed for it contains no flight instruction.

6.2 Checks are then made to validate the runway data. For example, phase names and aircraft types are compared with the phase data base and runway data base. Should these data be inconsistent, an error message is displayed and the run is terminated.

6.3 Subroutine GENLSR is called to compute the runway requirements. These requirements data are stored in the computer's memory and also written on the runway requirements data file, RUNWAY. The process continues until the runway requirements for all training phases have been computed. Next, the runway requirements data file, RUNWAY, is closed, and phase runway requirements printed. Control is then passed to PROGRAM PART2.

SUBROUTINE INPRWY

6.4 The purpose of Subroutine INPRWY is to read from the runway data file (RUNDAT) the phase runway data. Upon entry, all the runway data associated with a particular training phase are read into the computer's memory. Control then returns to PROGRAM LSR4.

SUBROUTINE GENRWY

6.5 Subroutine GENRWY generates training phase requirements for runways, OLFs, and air-to-ground target areas. Upon entry, monthly daylight hours, DH, are reduced by sortie length, SL, adjusted by monthly weather , WX, and runway downtime, DT, and averaged. The time factor, TIME, is the average hours per day that the runways may be utilized for pilot training.

6.6 Next the maximum number of daily sorties, SMAX, which can be launched and recovered, is computed as well as the required number of daily sorties, RS. Runway requirements are then developed by the ratio of required sorties to maximum sorties.

6.7 A similar process is used to compute OLF and air-to-ground target areas. Control then returns to PROGRAM LSR4.

6.8 A flow chart of LSR4 is shown in Figure 6. Table 15 contains the variable dictionary of PROGRAM LSR4; the program and subroutine dictionary is provided in Table 16. The program listing is shown in Table 17.

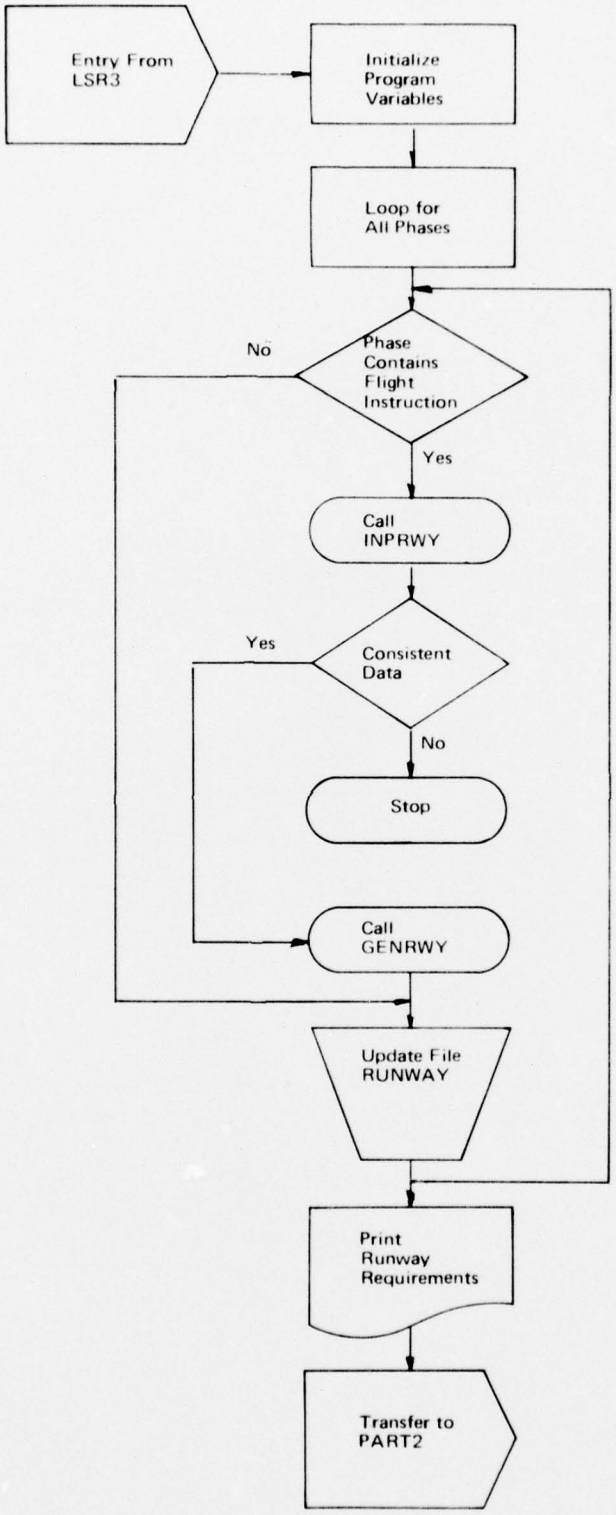


FIGURE 6. PROGRAM LSR4 FLOW CHART

a. Subroutine INPRWY

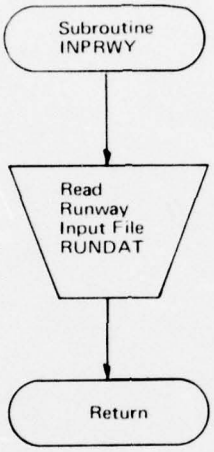


FIGURE 6 (Cont)

b. Subroutine GENRWY

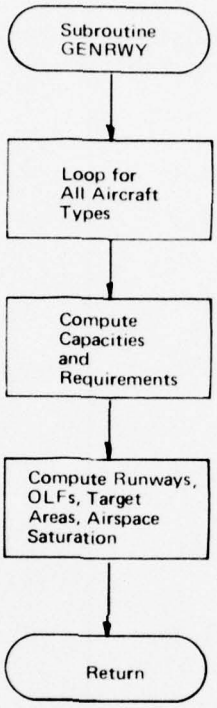


FIGURE 6. (Cont)

TABLE 15

PROGRAM LSR4 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Type	Description
Common	SWITCH	11	F	Permanent storage used by other IFRS computer programs
Common	NAME	25,3	A	Name of I^{th}* training phase
Common	NPLA	25,3	A	Name of J^{th}** type aircraft used in phase I
Common	IOPR	25,3	I	Runway operating procedure for J type aircraft in phase I
Common	SAS	25,3	F	Air space saturation factor for J^{th} aircraft in phase I
Common	OLF	25,3	F	OLF requirements for aircraft type J in phase I
Common	NAC	25	I	Number of aircraft types in phase I
Common	RUNP	25,3	F	Number of effective runways required for aircraft type J in phase I
Common	TARG	25,3	F	Air-to-ground target areas required for aircraft type J in phase I
Common	WX	3,12	F	Monthly weather for aircraft type I
Common	DH	12	F	Monthly daylight hours
Common	SP1	25	F	Used to space common (not used in program)
Common	SP2	25,27	F	Used to space common (not used in program)
Common	ICOMMA	1	A	Comma ","
Common	IBLANK	1	A	Space " "

* I refers to the column index.

** J refers to the row index.

TABLE 15 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	NO	1	A	No,"N"
Common	NYES	1	A	Yes,"Y"
Common	NY	1	I	No-Yes Switch NY = -1 for No response = 1 for Yes response
Common	NPH	1	I	Number of training phases
Common	IER	1	I	Error type switch
Common	LEVSLR	1	I	Level of complexity for LSR Generator
Common	IPH	1	I	Particular training phase
Common	WPY	1	F	Training weeks per year
Common	AFD	1	F	Annual number of flight training days
Common	KILL	1	I	Number of phases deleted in PROGRAM LSR1
Common	IID	1	I	Integer switch
Common	FID	1	I	Floating point switch
Common	KILLS	25	I	Deleted training phases
Common	SI	25	F	Student input for training phase I
Common	SO	25	F	Student output for training phase I
Common	NAMEP	3	A	Name of training phase in Runway Data File
Common	IAFT	3	A	Name of aircraft type I in Runway Data File
Common	SPS	3	F	Number of sorties per student for aircraft type I
Common	SL	3	F	Sortie length in aircraft type I
Common	TT	3	F	Time required to launch an I-type aircraft

TABLE 15 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	TL	3	F	Time required to recover an I-type aircraft
Common	AS	3	F	Number of I-type aircraft which will saturate the airspace
Common	ATAG	3	F	Number of touch-and-go landings each student must perform on an I-type aircraft
Common	PMR	3	F	Percent of the touch-and-go landings in an I-type aircraft conducted on the main runway
Common	TAR	3	F	Required number of passes over a target area by a student in an I-type aircraft
Common	TOT	3	F	Average time spent over a target area in an I-type aircraft
Common	DT	1	F	Percent of time a runway may not be used
Common	NACC	1	F	Number of aircraft
Common	DTO	1	F	Percent of time an OLF may not be used
LSR4	IL	1	I	Line counter
GENRWY	TIME	1	F	Average number of hours per day a runway may be utilized
GENRWY	TLC	1	F	Launch-recovery cycle
GENRWY	SMLC	1	F	Maximum number of daily sorties using a launch-recovery cycle
GENRWY	CYC	1	F	Maximum of launch and recovery time
GENRWY	SMTL	1	F	Maximum number of daily sorties using a batch launch and recovery cycle

TABLE 15 (Cont)

Location	Variable Name	Dimension	Type	Description
GENRWY	SMAX	1	F	Maximum of SMLC and SMTL
GENRWY	AIR	1	F	Maximum number of aircraft airborne at one time
GENRWY	ET	1	F	Time a main runway is used for touch-and-go landings
GENRWY	TGC	1	F	Daily capacity of OLF for touch-and-go landings
GENRWY	TGR	1	F	Daily touch-and-go requirements
GENRWY	OFR	1	F	Number of the daily touch-and-to landings to be conducted on an OLF
GENRWY	RS	1	F	Required number of daily sorties to be conducted
GENRWY	GUN	1	F	Daily number of air-to-ground target approaches
GENRWY	TART	1	F	Air-to-ground target area capacity

TABLE 16

PROGRAM LSR4 PROGRAM AND SUBROUTINE DICTIONARY

LSR4	Develops the program linkage for the computation of runway requirements. Prints runway requirements and develops the runway output file.
INPRWY	Reads the runway data file.
GENRWY	Generates the runway requirements.

TABLE 17

PROGRAM LSR4 LISTING

```
104       C0MM0N SWITCH(11)
124       C0MM0N NAME(25,3),NPLA(25,3),I0PR(25,3),SAS(25,3),0LF(25,3),
144      &NAC(25),RUNP(25,3),TARG(25,3),WX(3,12),DH(12),
164      &SP1(52),SP2(25,27)
184       C0MM0N IC0MMA,IBLANK,N0,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
204      &AFD,KILL,IID,FID,KILLS(25),SI(25),S0(25)
224        C0MM0N NAMEP(3),IAFT(3),SPS(3),SL(3),TT(3),TL(3),
244      &AS(3),ATAG(3),TAGT(3),PMR(3),TAR(3),T0T(3),DT,NACC
264      &,DT0
284       0PENFILE "RUNDAT"
304       0PENFILE "RUNWAY"
324       REWIND "RUNDAT"
344       REWIND "RUNWAY"
364       WRITE("RUNWAY",707)NPH
384       IL=1005
404       D0 10 I=1,NPH
424       D0 10 J=1,3
444       I0PR(I,J)=0
464       SAS(I,J)=0.0
484       0LF(I,J)=0.0
504       RUNP(I,J)=0.0
524    10 TARG(I,J)=0.0
544       D0 115 I=1,NPH
564       IPH=I
584       IF(NAC(I))100,100,20
604    20 CALL INPRWY
624       D0 40 J=1,3
644       IF(NAMEP(J)-NAME(I,J))30,40,30
664    30 PRINT 700,NAMEP,(NAME(I,K),K=1,3)
684       ST0P
704    40 C0NTINUE
724       IF(NAC(I)-NACC)50,60,50
744    50 PRINT 701,NACC,NAC(I),NAMEP
764       ST0P
784    60 C0NTINUE
804       D0 80 J=1,NACC
824       IF(IAFT(J)-NPLA(I,J))70,80,70
844    70 PRINT 702,NAMEP,IAFT(J),NPLA(I,J)
864       ST0P
```

TABLE 17 (Cont)

```
884    80 CALL GENRWY
904   100 WRITE("RUNWAY",708)IL,NPLA(I,1)
924       IL=IL+5
944       WRITE("RUNWAY",709)IL,(RUNP(I,J),J=1,3)
964       IL=IL+5
984       WRITE("RUNWAY",709)IL,(SAS(I,J),J=1,3)
1004       IL=IL+5
1024       WRITE("RUNWAY",709)IL,(ØLF(I,J),J=1,3)
1044  115 IL=IL+5
1064       PRINT 703
1084       DØ 200 I=1,NPH
1104       IF(NAC(I).LE.0)GØ TØ 200
1124       PRINT 704,(NAME(I,J),J=1,3),NPLA(I,1),RUNP(I,1),SAS(I,1),
1144      & ØLF(I,1),TARG(I,1)
1164       IF(NAC(I)-1)200,200,110
1184  110 K=NAC(I)
1204       DØ 120 J=2,K
1224  120 PRINT 705,NPLA(I,J),RUNP(I,J),SAS(I,J),ØLF(I,J),TARG(I,J)
1244       PRINT 706
1264  200 CØNTINUE
1284       CLØSEFILE "RUNWAY"
1304       CLØSEFILE "RUNDAT"
1324       CHAIN "PART2*"
1344  700 FØRMAT(" RUNWAY PHASE NAME ",3A4," DØES NØT MATCH PHAS
1364      &E NAME "3A4," REVISE AND RE-RUN")
1384  701 FØRMAT(" RUNWAY AIRCRAFT TYPES ØF",I3," DØES NØT MATCH"/
1404      &" PHASE TYPES ØF",I3," FØR PHASE ",3A4/" REVISE AND RE-RUN")
1424  702 FØRMAT(" FØR PHASE ",3A4," AIRCRAFT NAMES DØ NØT MATCH
1444     &PHASE AIRCRAFT NAMES ",A4,1H,,A4/" REVISE AND RE-RUN")
1464  703 FØRMAT(//18X,"A/C EFFECTIVE  AIRSPACE          TARGET"/
1484      &" TRAINING PHASE  TYPE  RUNWAYS  SATURATIØN   ØLF   AREAS")
1504  704 FØRMAT(1X,3A4,4X,A4,F8.3,F11.3,F8.3,F8.3)
1524  705 FØRMAT(17X,F8.3,F11.3,2F8.3)
1544  706 FØRMAT(1X)
1564  707 FØRMAT(5H1000 ,I3)
1584  708 FØRMAT(I4,1X,A4,"                B")
1604  709 FØRMAT(I4,1X,3E13.6)
1624       END
```

TABLE 17 (Cont)

a. Subroutine INPRWY

```
1644       SUBRØUTINE INPRWY
1664       CØMMØN SWITCH(11)
1684       CØMMØN NAME(25,3),NPLA(25,3),IØPR(25,3),SAS(25,3),ØLF(25,3),
1704      &NAC(25),RUNP(25,3),TARG(25,3),WX(3,12),DH(12),
1724      &SP1(52),SP2(25,27)
1744       CØMMØN ICØMMA,IBLANK,NØ,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
1764      &AFD,KILL,IID,FID,KILLS(25),SI(25),SØ(25)
1784         CØMMØN NAMEP(3),IAFT(3),SPS(3),SL(3),TT(3),TL(3),
1804      &AS(3),ATAG(3),TAGT(3),PMR(3),TAR(3),TØT(3),DT,NACC
1824      &,DTØ
1844       FILENAME RUN
1864       RUN="RUNDAT"
1884       READ(RUN,700)L,NACC,NAMEP,IAFT
1904       IF(NACC)10,10,20
1924    10 PRINT 701
1944       STØP
1964    20 READ(RUN,702)L,(DH(J),J=1,6)
1984       READ(RUN,702)L,(DH(J),J=7,12)
2004       READ(RUN,702)L,DT,DTØ
2024       DØ 30 I=1,NACC
2044       READ(RUN,702)L,(WX(I,J),J=1,6)
2064    30 READ(RUN,702)L,(WX(I,J),J=7,12)
2084       READ(RUN,702)L,SPS
2104       READ(RUN,702)L,SL
2124       READ(RUN,702)L,TT
2144       READ(RUN,702)L,TL
2164       READ(RUN,702)L,AS
2184       READ(RUN,702)L,ATAG
2204       READ(RUN,702)L,TAGT
2224       READ(RUN,702)L,PMR
2244       READ(RUN,702)L,TAR
2264       READ(RUN,702)L,TØT
2284    40 RETURN
2304   700 FØRMAT(2I4,6A4)
2324   701 FØRMAT(" RUNWAY DATA FILE IS INCØMPLETE -UPDATE AND RE-RUN")
2344   702 FØRMAT(V)
2364       END
```

TABLE 17 (Cont)

b. Subroutine GENRWY

```
2384       SUBRØUTINE GENRWY
2404       CØMMØN SWITCH(11)
2424       CØMMØN NAME(25,3),NPLA(25,3),IØPR(25,3),SAS(25,3),ØLF(25,3),
2444      &NAC(25),RUNP(25,3),TARG(25,3),WX(3,12),DH(12),
2464      &SP1(52),SP2(25,27)
2484       CØMMØN ICØMMA,IBLANK,NØ,NYES,NY,NPH,IER,LEVLSR,IPH,WPY,
2504      &AFD,KILL,IID,FID,KILLS(25),SI(25),SØ(25)
2524         CØMMØN NAMEP(3),IAFT(3),SPS(3),SL(3),TT(3),TL(3),
2544      &AS(3),ATAG(3),TAGT(3),PMR(3),TAR(3),TØT(3),DT,NACC
2564      &,DTØ
2584       DØ 300 I=1,NACC
2604       TIME=0.0
2624       DØ 10 J=1,12
2644    10 TIME=TIME+(DH(J)-SL(I))*WX(I,J)
2664       TIME=TIME*(1.0-DT)/12.0
2684       TLC=TT(I)+TL(I)
2704       SMLC=TIME/TLC
2724       CYC=TT(I)
2744       IF(CYC-TL(I))20,20,30
2764    20 CYC=TL(I)
2784    30 C=TIME/(2.0*SL(I))
2804       J=C
2824       C=J
2844       SMTL=C*SL(I)/CYC
2864       R=TIME-2.0*C*SL(I)
2884       E=SL(I)/CYC
2904       IF(R-SL(I))40,50,50
2924    40 E=R/CYC
2944    50 SMTL=SMTL+E
2964       IF(SMLC-SMTL)60,60,70
2984    60 SMAX=SMTL
3004       IØPR(IPH,I)=2
3024       AIR=SL(I)/CYC
3044       GØ TØ 80
3064    70 SMAX=SMLC
3084       IØPR(IPH,I)=1
3104       AIR=SL(I)/TLC
```

TABLE 17 (Cont)

b. Subroutine GENRWY (Cont)

```
3124    80 ET=0.
3144       IF(AIR.GT.AS(I))GØ TØ 200
3164    85 SAS(IPH,I)=AIR/AS(I)
3184       IF(ATAG(I))100,100,90
3204    90 TGC=TIME*(1.-DTØ)/TAGT(I)
3224       TGR=SØ(IPH)*ATAG(I)/AFD
3244       ØFR=(1.0-PMR(I))*TGR
3264       ØLF(IPH,I)=ØFR/TGC
3284       ET=(TGR-ØFR)*(1.0-DTØ)/TGC
3304   100 RS=SØ(IPH)*SPS(I)/AFD
3324       RUNP(IPH,I)=ET + RS/SMAX
3344       IF(TAR(I))290,290,110
3364   110 GUN=SØ(IPH)*TAR(I)/AFD
3384       TART=(TIME-SL(I))/TØT(I)
3404       TARG(IPH,I)=GUN/TART
3424       GØ TØ 300
3444   200 IF(IØPR(IPH,I).EQ.2)GØ TØ 70
3464       C=TIME/SL(I)
3484       J=C
3504       D=J
3524       C=C-D
3544       T=C/TLC
3564       IF(T-AS(I))220,220,210
3584   210 T=AS(I)
3604   220 SMAX=D*AS(I)+T
3624       AIR=AS(I)
3644       IØPR(IPH,I)=3
3664       GØ TØ 85
3684   290 SAS(IPH,I)=SAS(IPH,I)*RUNP(IPH,I)
3704   300 CØNTINUE
3724       RETURN
3744       END
```

VII. PROGRAM PART2

PROGRAM DESCRIPTION

7.1 PROGRAM PART2 either accepts control from the LSR Generator or is entered directly from the time-sharing terminal. Its purpose is to read several files: the Base Data File; BASED*; the Aircraft Data File, ACDAT*; and, in the event that control is being transferred back to PART2 after it has been run previously, the program reads one or both of the temporary data files, RETURN and RETURN1.

7.2 Since PROGRAMS LSRM through LSR4 have a different set of common variables than PROGRAMS PART2 through PARTY, with the exception of the first 11 locations, certain variables in common in PART2 have to be initialized. This is done upon entering PART2. The array ISWTCH within the first 11 common locations acts as the sole internal link among PROGRAMS LSRM through LSR4 and PROGRAMS PART2 through PARTY. The values of the switches in ISWTCH convey such information as whether the run is currently in a multi-year mode (ISWTCH(10) = 1), or in the constrained resource-LSR mode originating from PART8 (ISWTCH(10) = 2). ISWTCH(10) = 0 the first time PROGRAM PART2 is entered. If this occurs, the date is set at 1970 and ISWTCH(6); the year counter is set to 1.

7.3 Both the Base Data File (BASED*) and the Aircraft Data File (ACDAT*) are then read. If this is the first time through, control transfers to PART3. If this is a multi-year run, i.e., ISWTCH(10) $\neq$ 0, and the model is beyond the first year computations, i.e., ISWTCH(6) $\neq$ 1, then the updated asset position and phase to base assignment from the previous year (file RETURN) are read.

7.4 If, at some previous point, the option has been made to return to the LSR Generator from one of the cost models to constrain output, i.e., ISWTCH (10) = 2, the phase to base assignment (the last one established before the transfer

to the LSR Generator) is read from file RETURN1. Following this, control is transferred to PART3.

7.5 A flow chart of PROGRAM PART2 is shown in Figure 7. Table 18 contains the variable dictionary of PROGRAM PART2; the program and subroutine dictionary is provided in Table 19. The program listing is shown in Table 20.

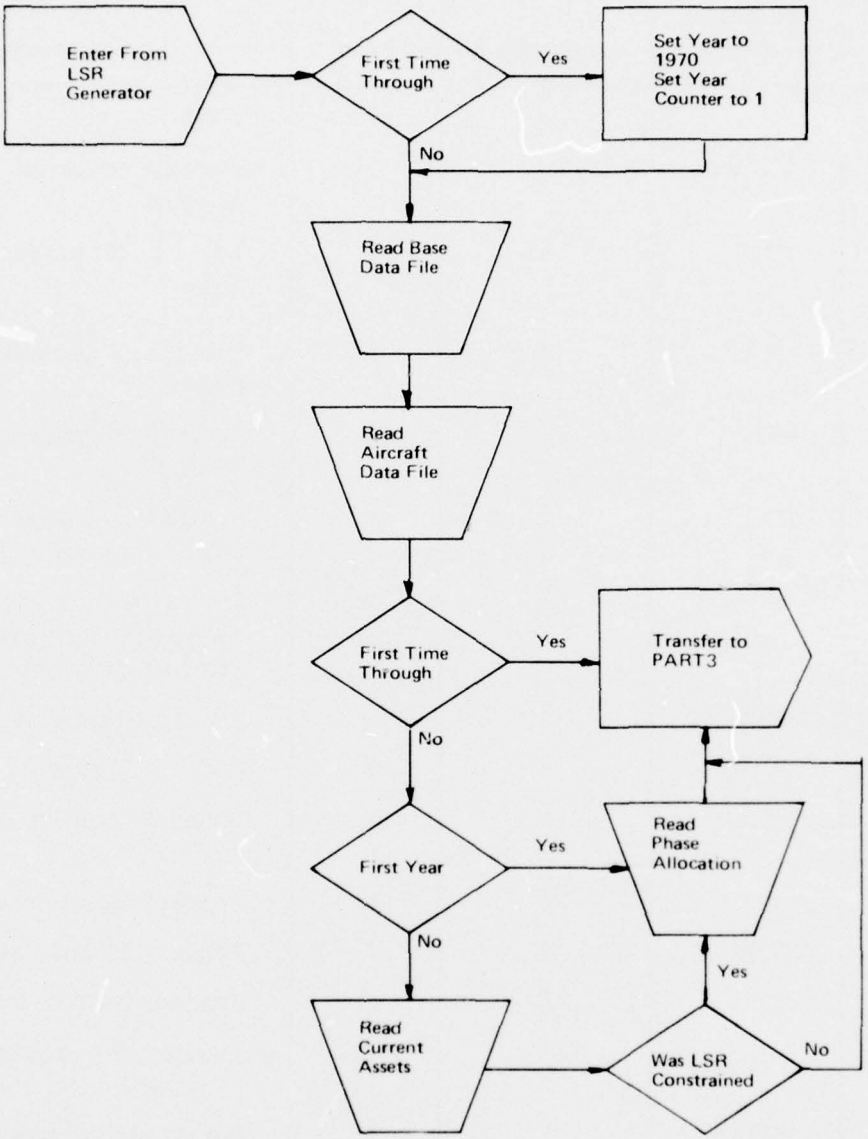


FIGURE 7. PROGRAM PART2 FLOW CHART

TABLE 18

PROGRAM PART2 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Type	Description
Common	IYEAR	1	I	Date
Common	ACREQ	9,21	F	Aircraft required of type J at base I*
Common	TBAS	9	F	Total base personnel at base I
Common	TNAS	9	F	Total NAS personnel at base I
Common	BPH	9,25	F	Percent of phase J at base I
Common	ASH	25,3	F	Academic student hours in phase I for type J instruction
Common	ACFH	9,15	F	Annual aircraft flight hours for type J aircraft (training aircraft only) at base I
Common	TOFF	9	F	Total officers at base I
Common	TENL	9	F	Total enlisted at base I
Common	PNASE	9	F	Total NAS enlisted at base I
Common	SI	25	F	Student input into phase I
Common	TCIV	9	F	Total civilians at base I
Common	SO	25	F	Student output from phase I
Common	FUREQ	9,3	F	Fuel type J required annually at base I
Common	PHPER	9,5	F	Personnel at base I of type J: J = 1 for phase officers and students, J = 2 for phase enlisted, J = 3 for NAS officers, J = 4 for NAS enlisted, J = 5 for NAS civilian
Common	NBUSE	9	I	Base usage code: 0 = not used 1 = used

* I always denotes the first array index, J the second, K the third.

TABLE 18 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	RW	25,3,3	F	Presently unused
Common	IACT	25,3	A	Aircraft type required for phase I, instruction type J
Common	ACNO1	25,3	F	Number of aircraft required for phase I, instruction type J
Common	TOFF1	25	F	Total officers in phase I
Common	EMTI	25	F	Total enlisted men in phase I
Common	IATYPE	21	A	Aircraft name of type I aircraft
Common	ACA	21	F	Aircraft parking apron variable A for type I aircraft
Common	ACB	21	F	Aircraft parking apron variable B for type I aircraft
Common	ACC	21	F	Aircraft parking apron variable C for type I aircraft
Common	ACD	21	F	Aircraft parking apron variable D for type I aircraft
Common	AHM	21	F	Number of aircraft of type I per hangar module
Common	ACM	21	F	Number of aircraft of type I per crew and equipment module
Common	ASM1	21	F	Number of aircraft of type I per basic shop module
Common	ASM2	21	F	Number of aircraft of type I per supplementary shop module
Common	A	21,3	F	Warehouse space type J required for aircraft type I

TABLE 18 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	RNWYL	21	F	Runway length required for type I aircraft
Common	RLOAD	21	F	Runway load factor required for type I aircraft
Common	COMP	21	F	Runway composition factor required for type I aircraft: 1 = concrete, 2 = asphalt
Common	FLCST	21	F	Flyaway cost for type I aircraft
Common	AOM	21	F	Aircraft operating and maintenance cost per flight hour for type I aircraft
Common	CNAAC	21	F	Aircraft available to CNATRA, type I
Common	NASNAM	9	A	NAS name, base I (first 4 letters of name)
Common	AD	9	F	Parking apron depth at base I
Common	PF	9,3	F	Days of ready fuel storage required at base I for type J fuel
Common	EL	9,3	F	Fuel loss factor for base I, type J fuel
Common	CU	9	F	Annual classroom utilization, base I
Common	IBED	9	I	Dispensary code by base: 1 = with beds, 0 = without
Common	PEE	9	F	Percent of enlisted eligible for family housing at base I
Common	PRE	9	F	Percent of eligible enlisted requiring family housing at base I
Common	PO	9	F	Percent of officers requiring family housing at base I

TABLE 18 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	PS	9	F	Percent of students requiring family housing at base I
Common	PIE	9	F	Percent of ineligible enlisted requiring family housing at base I
Common	TS	9	F	Annual tenant students at base I
Common	TH	9	F	Annual tenant classroom utilization at base I
Common	TNOFF	9	F	Tenant officers at base I
Common	TNENL	9	F	Tenant enlisted at base I
Common	TNCIV	9	F	Tenant civilians at base I
Common	ATCF	9	F	Altitude temperature correction factor for runways at base I
Common	WR	9,2	F	Wind rose data for base I type J runways: J = 1 for primary, J = 2 for crosswind
Common	TENAC	9,6	F	Tenant aircraft type J at base I
Common	PERFAC	9	F	Percent of total base personnel at base I requiring administrative office space
Common	EMES	9	F	Percent of enlisted men barrack capacity utilizing mess halls
Common	FACOST	50,6	F	Cost factor J for facility I
Common	FAPW	6	F	Maximum floor area for public works shop for personnel level I
Common	AP	4,3	F	Square feet of type J storage required for base size I

TABLE 18 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	GWTAB	3	F	Maximum number of base personnel for base size I (warehouse computations)
Common	FAMESS	7,2	F	FAMESS(I,1) = max. total mess capacity I; FAMESS (I,2) = sq ft per person required for mess capacity I
Common	EXCH	10,2	F	EXCH(I,1) = base military strength level I for exchange calculations; EXCH(I,2) = floor area required for exchange for base strength I
Common	FAEM	8,2	F	FAEM(I,1) = cutoff enlisted strength level I for service club calculations; FAEM (I,2) = floor area required for EM service club for enlisted level I
Common	TANKS	15	F	Tank sizes available for ready fuel storage low to high
Common	TAXITO	3	F	Taxiway turnoffs required for runway length I
Common	ICODES	50	I	Category code associated with I^{th} facility
Common	IDES	50,3	A	Facility description (12 characters) for facility I
Common	RPI	50,9,2	F	Facility I inventory at base J for K type: K = 1 for standard, K = 2 for substandard
Common	IUNITS	50	A	Unit of measure for facility I (2 characters)
Common	XRPI1	9,10,4	F	Available runway J description (4 items-indexed K) at base I

TABLE 18 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	XRPI2	3,9	F	Ready fuel storage type I available at base J
Common	BR	50,9	F	Required amount of facility I at base J
Common	XBR1	9,10,4	F	Required runway J description (4 items-indexed K) at base I
Common	XBR2	3,9	F	Ready fuel storage type I required at base J
Common	DEF	50,9	F	Facility I deficiency at base J
Common	XDEF2	9	F	Taxiway deficiency at base J
Common	XDEF3	2,9	F	I = 1 for runway lighting deficit (linear feet) at base J; I = 2 for number of approach lighting system deficient at base J
Common	XDEF4	3,15,9	F	Number of tanks of size J (see TANKS) required at base K for fuel type I
Common	TEX	50,9	F	Total excess of facility I at base J
Common	NCAT	1	I	Number of facilities considered
Common	IYES	1	A	Character "Y" for recognizing Yes responses
Common	NO	1	A	Character "N" for recognizing No responses
Common	ICOM	1	A	Character "," for validating input format
Common	GTOTAL	1	F	Total systems cost
Common	NPH	1	I	Number of phases of training

TABLE 18 (Cont)

Location	Variable Name	Dimension	Type	Description
Common	ISWTCH	10	I	ISWTCH(1) to ISWTCH(5) unused ISWTCH(6) year counter for multi-year runs ISWTCH(7) = 1 for restart at PART6 ISWTCH(8) = 1 for reallocation of phases ISWTCH(9) = level of print detail ISWTCH(10) = 1 for multi-year return to LSR = 2 for constrained LSR option = 0 otherwise
PART2	LINE	1	I	Dummy variable

TABLE 19

PROGRAM PART2 PROGRAM AND SUBROUTINE DICTIONARY

PART2	Reads Base Data File, Aircraft Data File, and, if needed, Return Files

TABLE 20

PROGRAM PART2 LISTING

```
1002      COMMON IYEAR,ISWTCH(10)
1022      COMMON ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
1042     &ACFH(9,15),TOFF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
1062     &SO(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
1082     &IACT(25,3),ACNO1(25,3),TOFF1(25),EMT1(25)
1102      COMMON IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
1122     &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
1142     &RLOAD(21),COMP(21),FLCST(21),AOM(21),CNAAC(21)
1162      COMMON NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
1182     &PRE(9),PO(9),PS(9),PIE(9),TS(9),TH(9),TNOFF(9),TNENL(9),
1202     &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
1222      COMMON FACOST(50,6)
1242      COMMON FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
1262     &FAEM(8,2),TANKS(15),TAXITO(3)
1282      COMMON ICODES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
1302     &XRPI1(9,10,4),XRPI2(3,9)
1322      COMMON BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
1342     &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
1362     &NCAT,IYES,NO,ICOM,GTOTAL,NPH
1382      ALPHA ICOM,IYES,NO,IATYPE
1402      DO 1 I=1,21
1422      DO 1 J=1,9
1442    1 ACREQ(J,I)=0.
1462      ICOM=","
1482      NO="N"
1502      IYES="Y"
1522      IF(ISWTCH(10).EQ.0)GO TO 20
1542   15 OPENFILE "BASED*"
1562      REWIND "BASED*"
1582      DO 18 I=1,9
1602      READ("BASED*",600)NASNAM(I)
1622      READ("BASED*",602)LINE,AD(I)
1642      READ("BASED*",602)LINE,(PF(I,J),J=1,3),(EL(I,K),K=1,3)
1662      READ("BASED*",602)LINE,CU(I),TH(I),TS(I)
1682      READ("BASED*",602)LINE,TNOFF(I),TNENL(I),TNCIV(I)
1702      READ("BASED*",602)LINE,PEE(I),PRE(I),PO(I),PS(I),PIE(I)
1722      READ("BASED*",602)LINE,EMES(I),IBED(I),PERFAC(I)
1742      READ("BASED*",602)LINE,ATCF(I),(WR(I,J),J=1,2)
1762      READ("BASED*",602)LINE,(TENAC(I,J),J=1,6)
1782   18 CONTINUE
1802      CLOSEFILE "BASED*"
1822      OPENFILE "ACDAT*"
1842      REWIND "ACDAT*"
```

TABLE 20 (Cont)

```
1862        DØ 19 I=1,21
1882        READ("ACDAT*",600)IATYPE(I)
1902        READ("ACDAT*",606)LINE,ACA(I),ACB(I),ACC(I),ACD(I)
1922        READ("ACDAT*",606)LINE,AHM(I),ACM(I),ASM1(I),ASM2(I)
1942        READ("ACDAT*",606)LINE,(A(I,J),J=1,3)
1962        READ("ACDAT*",606)LINE,RNWYL(I),RLØAD(I),CØMP(I)
1982        READ("ACDAT*",606)LINE,FLCST(I),AØM(I)
2002     19 READ("ACDAT*",606)LINE,CNAAC(I)
2022        CLØSEFILE "ACDAT*"
2042        IF(ISWTCH(10).EQ.0)GØ TØ 30
2062        IF(ISWTCH(6).EQ.1)GØ TØ 195
2082        ØPENFILE "RETURN"
2102        REWIND "RETURN"
2122        READ("RETURN",601)ICØDES,NBUSE
2142        READ("RETURN",603)IDES,IUNITS
2162        READ("RETURN",604)RPI,XRPI1,XRPI2,FACØST,BPH,CNAAC
2182        CLØSEFILE "RETURN"
2202        IF(ISWTCH(10).EQ.2)GØ TØ 195
2222        GØ TØ 30
2242   195  ØPENFILE "RETURN1"
2262        REWIND "RETURN1"
2282        READ("RETURN1",604)BPH
2302        READ("RETURN1",601)NBUSE
2322        GØ TØ 30
2342     20 IYEAR=1970
2362        ISWTCH(6)=1
2382        GØ TØ 15
2402     30 CHAIN "PART3*"
2422    600 FØRMAT(5XA4)
2442    601 FØRMAT(8I8)
2462    602 FØRMAT(V)
2482    603 FØRMAT(15A4)
2502    604 FØRMAT(5E13.6)
2522    606 FØRMAT(V)
2542        END
```

VIII. PROGRAM PART3

PROGRAM DESCRIPTION

8.1 PROGRAM PART3 [1] records the phase to base assignments, reads the file LSROUT containing the personnel, aircraft, and fuel requirements generated by the LSR Generator, and apportions these resources to their respective bases.

8.2 Upon entering PART3, three alpha variables are initialized so that the fuel types required, as developed in the LSR Generator, can be recognized. Next the tenant data previously recorded in PART2 are extracted from common and placed in array TDATA for each base.

8.3 The file LSROUT, created by the LSR Generator, is then read. If this is not the first time through this program, the user is given the option of retaining his previous phase to base assignment.

8.4 The description of how to allocate phases is printed, unless this is a reallocation of phases (ISWTCH (8) = 1). Following this, each phase allocation is entered from the terminal. Each allocation is thoroughly checked for validity. The phase number is compared against the maximum phases currently included and the base code and percentage value are checked.

8.5 After the user indicates termination of input (i.e., 0, 0), a further check is made to see if each phase has been fully allocated, but not overallocated. If a phase is overallocated, all previous allocations for that phase are deleted and a request is generated to the user for a new phase assignment. If a phase is underallocated, the user is asked simply for another assignment.

---

[1] This program is essentially the Base Loading Submodel discussed in Volumes I and II.

8.6 Once all phases have been properly assigned, the variable NBUSE (I) is set to 1 or 0 depending on whether base I is or is not in use. Next, the user specifies whether he wishes to see a detailed or summary version of the resulting base loading data.

8.7 Personnel figures are then aggregated and printed, either in summary or detailed form. Next, aircraft types and requirements are aggregated by base and stored in ATYPE and PLREQ, respectively. Also at this time, total annual flight hours, by aircraft type (HRSREQ), are aggregated for computing O&M costs later in PART9. Tenant aircraft are also included in the aggregation.

8.8 Fuel names and requirements are aggregated and stored by fuel type in FTYPE and GAREQ. If the fuel type generated by the LSR model is not recognizable, i.e., the first character of the name is not J, A, or H, an error message is printed at the terminal.

8.9 The fuel requirements and aircraft requirements are then converted from the internal arrays to the common arrays FUREQ and ACREQ, respectively. The printouts (summary or detailed) are displayed and the user is then given the option to reallocate phases. If he does, the whole procedure is repeated, following the deletion of the previous allocations. If not, then control transfers to PART4.

8.10 A flow chart of PROGRAM PART3 is shown in Figure 8. Table 21 contains the variable dictionary of PROGRAM PART3; the program and subroutine dictionary is provided in Table 22. The program listing is shown in Table 23.

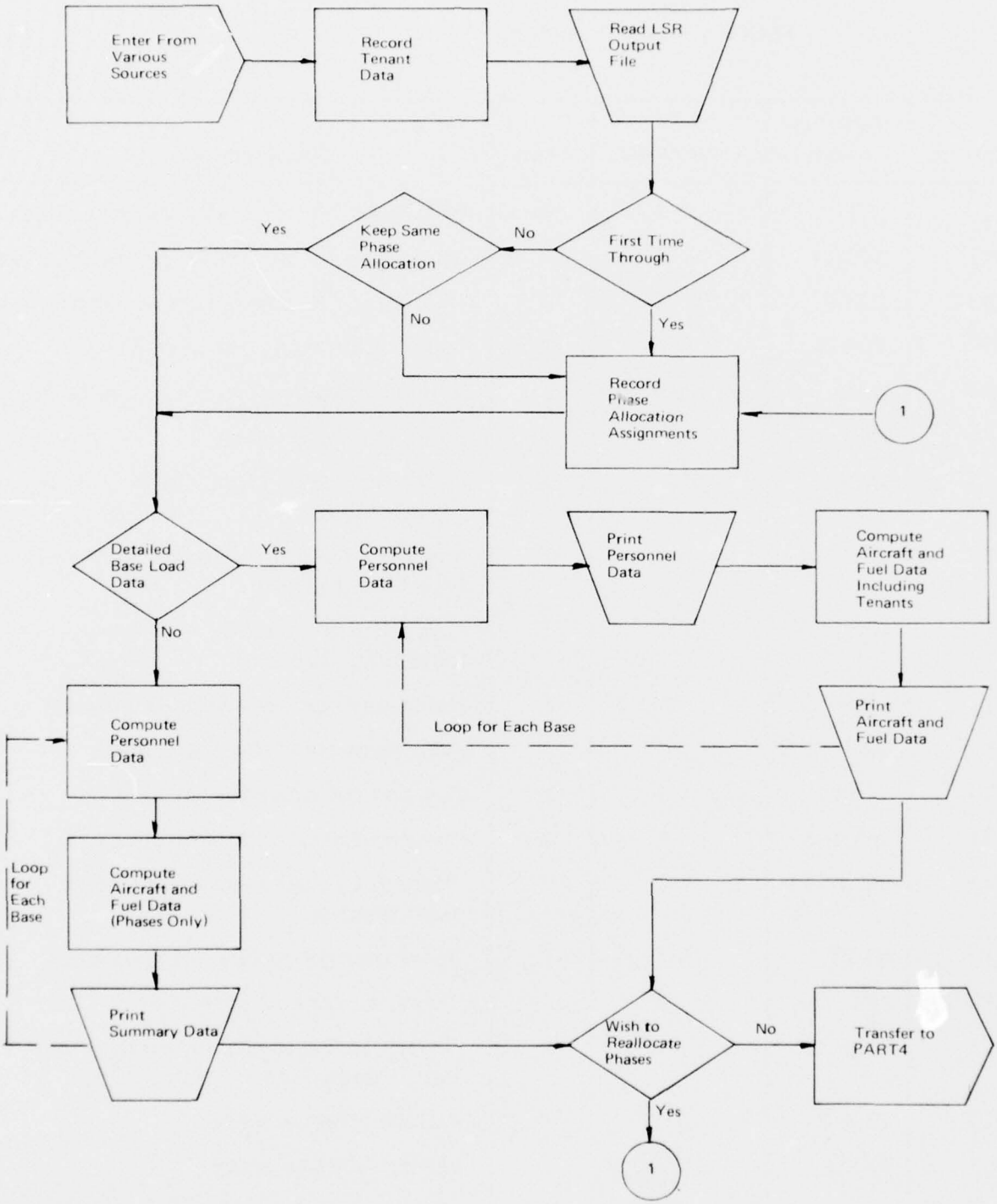


FIGURE 8. PROGRAM PART3 FLOW CHART

TABLE 21

PROGRAM PART3 VARIABLE DICTIONARY*

Location	Variable Name	Dimension	Type	Description
PART3	JET	1	A	Character "J" for fuel type determination
PART3	AVGAS	1	A	Character "A" for fuel type determination
PART3	HELO	1	A	Character "H" for fuel type determination
PART3	TDATA	9,4	F	Tenant data type J at base I
PART3	NAC	25	I	Number of aircraft types in phase I
PART3	SL	25	F	Student load in phase I
PART3	IAFT	25,3	A	Fuel type required for phase I, type J instruction
PART3	BF1	25,3	F	Annual bulk fuel required in phase I, type J instruction
PART3	BCFH	25,3	F	Flight hours annually in phase I, instruction type J
PART3	IPH	1	I	Phase number input from terminal
PART3	AA	1	A	Base name input from terminal
PART3	ICOM1	1	A	Common for input format check
PART3	ICOM2	1	A	Common for input format check
PART3	PCT	1	F	Percent of phase allocated (input from terminal)
PART3	TPCT	25	F	Total percent of phase I allocated
PART3	IOP	1	A	Terminal Yes-No response
PART3	NODETL	1	I	Switch indicator for detailed or no-detail base loading data
PART3	OOUT	25	F	Output storage array
PART3	OUT1	25,3	F	Output storage array

* For variables in common, see PROGRAM PART2 variable dictionary, Table 18.

TABLE 21 (Cont)

Location	Variable Name	Dimension	Type	Description
PART3	NAME	25,3	A	Phase I name (12 characters)
PART3	SUM1-4	4	F	Personnel partial sum accumulators
PART3	PNAS	4	F	NAS personnel, type I
PART3	XBAS	20	F	Total base personnel, type I
PART3	ATYPE	20	A	Aircraft name table
PART3	PLREQ	20	F	Number of aircraft of type I required
PART3	HRSREQ	20	F	Annual flight hours for type I aircraft
PART3	FTYPE	20	A	Fuel type for accumulating over phases
PART3	GAREQ	20	F	Gallons of fuel required of type I
PART3	JTYPE	20	A	Altered fuel names (first character only of original name)
PART3	WAG	1	F	Gallons of fuel required in millions

TABLE 22

PROGRAM PART3 PROGRAM AND SUBROUTINE DICTIONARY

PART3	Reads LSR output, accepts phase to base assignments,and allocates personnel, aircraft,and fuel requirements by base

TABLE 23

PROGRAM PART3 LISTING

```
      COMMON IYEAR,ISWTCH(10)
      COMMON ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
     &ACFH(9,15),TOFF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
     &SO(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
     &IACT(25,3),ACNO1(25,3),TOFF1(25),EMT1(25)
      COMMON IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
     &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
     &RLOAD(21),COMP(21),FLCST(21),AOM(21),CNAAC(21)
      COMMON NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
     &PRE(9),PO(9),PS(9),PIE(9),TS(9),TH(9),TNOFF(9),TNENL(9),
     &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
      COMMON FACOST(50,6)
      COMMON FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
     &FAEM(8,2),TANKS(15),TAXITO(3)
      COMMON ICODES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
     &XRPI1(9,10,4),XRPI2(3,9)
      COMMON BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
     &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
     &NCAT,IYES,NO,ICOM,GTOTAL,NPH
      DIMENSION OOUT(25),TPCT(25),PNAS(4),OUT1(25,3),
     &TDATA(9,4),ATYPE(20),FTYPE(20),PLREQ(20),
     &SL(25),GAREQ(20),BCFH(25,3),
     &BF1(25,3),FUEL(25,3),NAME(25,3),NAC(25)
     &,IAFT(25,3),JTYPE(20),XBAS(4),HRSREQ(20)
      ALPHA AA,ICOM,ICOM1,ICOM2,IYES,NO,ATYPE,FTYPE,
     &NASNAM,NAME,IACT,IAFT,JET,AVGAS,HELO,IOP,JTYPE
     &,IATYPE
      JET="J"
      AVGAS="A"
      HELO="H"
      DO 1000 I=1,9
      TDATA(I,1)=TNOFF(I)
      TDATA(I,2)=TNENL(I)
      TDATA(I,3)=TNCIV(I)
1000  TDATA(I,4)=TDATA(I,1)+TDATA(I,2)+TDATA(I,3)
      OPENFILE "LSROUT"
      REWIND "LSROUT"
      READ("LSROUT",651)NPH
      DO 2 I=1,NPH
      READ("LSROUT",652)(NAME(I,J),J=1,3),NAC(I)
      READ("LSROUT",653)SI(I),SO(I),SL(I),TOFF1(I),EMT1(I)
      READ("LSROUT",629)(IACT(I,J),J=1,3),(IAFT(I,J1),J1=1,3)
      READ("LSROUT",630)(ACNO1(I,J),J=1,3)
```

TABLE 23 (Cont)

```
1863       READ("LSR0UT",630)(BF1(I,J),J=1,3)
1883       READ("LSR0UT",630)(ASH(I,J),J=1,3)
1903       READ("LSR0UT",630)(BCFH(I,J),J=1,3)
1923     2 C0NTINUE
1943       CL0SEFILE "LSR0UT"
1963       IF(ISWTCH(10).GT.0)G0 T0 500
1983     4 D0 5 I=1,25
2003       D0 5 J=1,9
2023     5 BPH(J,I)=0.
2043       IF(ISWTCH(8).EQ.0)G0 T0 9
2063       ISWTCH(8)=0
2083       PRINT,"TYPE FIRST BASE ASSIGNMENT"
2103       G0 T0 10
2123     9 PRINT 600
2143    10 INPUT 601,IPH,IC0M1,AA,IC0M2,PCT
2163       IF(IPH)20,130,40
2183    40 IF(IPH.GT.NPH)G0 T0 25
2203       IF(IC0M1.EQ.IC0M)G0 T0 30
2223    20 PRINT 602
2243       G0 T0 10
2263    25 PRINT," N0 SUCH PHASE--RETYPE"
2283       G0 T0 10
2303    30 IF(IC0M2.NE.IC0M)G0 T0 20
2323    70 D0 90 I=1,9
2343       IF(AA.NE.NASNAM(I))G0 T0 90
2363       K=I
2383       G0 T0 100
2403    90 C0NTINUE
2423       PRINT 604
2443       G0 T0 10
2463   100 IF(PCT-1.)120,120,110
2483   110 PRINT 605,PCT
2503       G0 T0 10
2523   120 BPH(K,IPH)=PCT
2543   125 PRINT 606
2563       G0 T0 10
2583   130 D0 140 I=1,25
2603       TPCT(I)=0.
2623       D0 140 J=1,9
2643   140 TPCT(I)=TPCT(I)+BPH(J,I)
2663       D0 160 I=1,NPH
2683       IF (TPCT(I)-.995)150,145,145
2703   145 IF(TPCT(I)-1.005)160,160,147
```

TABLE 23 (Cont)

```
2723   147 K=I
2743       GØ TØ 165
2763   150 K=I
2783       GØ TØ 170
2803   160 CØNTINUE
2823       GØ TØ 190
2843   165 PRINT 625,K
2863       DØ 167 I=1,9
2883   167 BPH(I,K)=0.
2903       GØ TØ 125
2923   170 PRINT 607,K
2943       GØ TØ 125
2963   190 DØ 198 I=1,9
2983       TEMP=0.
3003       DØ 195 J=1,25
3023   195 TEMP=TEMP+BPH(I,J)
3043       IF(TEMP-.01)196,196,197
3063   196 NBUSE(I)=0
3083       GØ TØ 198
3103   197 NBUSE(I)=1
3123   198 CØNTINUE
3143   520 PRINT,"DØ YØU WANT DETAILED BASE LØADING DATA (Y,N)"
3163       NØDETL=0
3183   200 INPUT,IØP
3203       IF(IØP.EQ.IYES)GØ TØ 210
3223       IF(IØP.EQ.NØ)GØ TØ 205
3243       PRINT,"INVALID REPLY--TRY AGAIN"
3263       GØ TØ 200
3283   205 NØDETL=1
3303       PRINT 665
3323   210 DØ 400 IB=1,9
3343       IF(NBUSE(IB))400,400,265
3363   265 K=0
3383       IF(NØDETL.EQ.1)GØ TØ 267
3403       PRINT 715,NASNAM(IB)
3423   267 DØ 280 I=1,NPH
3443       C=BPH(IB,I)
3463       IF(C-.01)280,280,270
3483   270 K=K+1
3503       ØØUT(K)=C*SL(I)
3523       ØUT1(K,1)=C*TØFF1(I)
3543       ØUT1(K,2)=C*EMT1(I)
3563       ØUT1(K,3)=ØUT1(K,1)+ØUT1(K,2)+ØØUT(K)
```

TABLE 23 (Cont)

```
3583        IF(NØDETL.EQ.1)GØ TØ 280
3603        PRINT 716,(NAME(I,J),J=1,3),ØØUT(K),(ØUT1(K,J),J=1,3)
3623   280  CØNTINUE
3643        SUM1=0.
3663        SUM2=0.
3683        SUM3=0.
3703        SUM4=0.0
3723        DØ 284 I=1,K
3743        SUM1=SUM1+ØUT1(I,1)
3763        SUM2=SUM2+ØUT1(I,2)
3783        SUM4=SUM4+ØØUT(I)
3803   284  SUM3=SUM3+ØUT1(I,3)
3823        TSTU(IB)=SUM4
3843        PHPER(IB,1)=SUM1+SUM4
3863        PHPER(IB,2)=SUM2
3883        IF(NØDETL.EQ.1)GØ TØ 2084
3903        PRINT 718,SUM4,SUM1,SUM2,SUM3
3923        PRINT 719,(TDATA(IB,J),J=1,4)
3943  2084  PNAS(4)=518.4+.259*(TDATA(IB,4)+SUM3)
3963        PNAS(2)=407.9+.0939*(TDATA(IB,4)+SUM3)
3983        PNAS(1)=19.23+.1765*(TDATA(IB,1)+SUM1)
4003        PNASE(IB)=PNAS(2)
4023        TNAS(IB)=PNAS(4)
4043        PNAS(3)=PNAS(4)-PNAS(1)-PNAS(2)
4063        PHPER(IB,3)=PNAS(1)
4083        PHPER(IB,4)=PNAS(2)
4103        PHPER(IB,5)=PNAS(3)
4123        XBAS(1)=PNAS(1)+TDATA(IB,1)+SUM1
4143        XBAS(2)=PNAS(2)+TDATA(IB,2)+SUM2
4163        XBAS(3)=PNAS(3)+TDATA(IB,3)
4183        TCIV(IB)=XBAS(3)
4203        TØFF(IB)=XBAS(1)
4223        TENL(IB)=XBAS(2)
4243        XBAS(4)=PNAS(4)+TDATA(IB,4)+SUM3
4263        TBAS(IB)=XBAS(4)
4283        IF(NØDETL.EQ.1)GØ TØ 2085
4303        PRINT 720,(PNAS(I),I=1,4),(XBAS(J),J=1,4)
4323  2085  K=0
4343        NF=1
4363        DØ 300 I=1,NPH
4383        IF(NAC(I).EQ.0)GØ TØ 300
4403        C=BPH(IB,I)
4423        JLØW=1
```

TABLE 23 (Cont)

```
4443        IF(C-.01)300,300,285
4463   285  IF(NF-1)286,286,288
4483   286  K=K+1
4503        ATYPE(K)=IACT(I,1)
4523        PLREQ(K)=ACN01(I,1)*C
4543        HRSREQ(K)=BCFH(I,1)*C
4563        NF=2
4583        IF(NAC(I)-1)300,300,287
4603   287  JL0W=2
4623   288  JHI=NAC(I)
4643        D0 293 J=JL0W,JHI
4663        L=1
4683   289  IF(IACT(I,J).NE.ATYPE(L))G0 T0 291
4703        PLREQ(L)=PLREQ(L)+ACN01(I,J)*C
4723        HRSREQ(L)=HRSREQ(L)+BCFH(I,J)*C
4743        G0 T0 293
4763   291  L=L+1
4783        IF(L-K)289,289,292
4803   292  K=K+1
4823        ATYPE(K)=IACT(I,J)
4843        PLREQ(K)=ACN01(I,J)*C
4863        HRSREQ(K)=BCFH(I,J)*C
4883   293  C0NTINUE
4903   300  C0NTINUE
4923        L1=K
4943        IF(K.EQ.0)N0AC=1
4963        D0 301 I=16,21
4983        IX=I-15
5003        IF(TENAC(IB,IX).LT..01)G0 T0 301
5023        K=K+1
5043        ATYPE(K)=IATYPE(I)
5063        PLREQ(K)=TENAC(IB,IX)
5083   301  C0NTINUE
5103        KF=1
5123        NF=1
5143        D0 350 I=1,NPH
5163        C=BPH(IB,I)
5183        JL0W=1
5203        IF(NAC(I).EQ.0)G0 T0 350
5223        IF(C-.01)350,350,335
5243   335  IF(NF-1)336,336,338
5263   336  FTYPE(KF)=IAFT(I,1)
5283        GAREQ(KF)=BF1(I,1)*C
```

TABLE 23 (Cont)

```
5303        NF=2
5323        IF(NAC(I)-1)350,350,337
5343   337  JL0W=2
5363   338  JHI=NAC(I)
5383        D0 343 J=JL0W,JHI
5403        L=1
5423   339  IF(IAFT(I,J).NE.FTYPE(L))G0 T0 341
5443        GAREQ(L)=GAREQ(L)+BF1(I,J)*C
5463        G0 T0 343
5483   341  L=L+1
5503        IF(L-KF)339,339,342
5523   342  KF=KF+1
5543        FTYPE(KF)=IAFT(I,J)
5563        GAREQ(KF)=BF1(I,J)*C
5583   343  C0NTINUE
5603   350  C0NTINUE
5623        0PENFILE "SCR1"
5643        REWIND "SCR1"
5663        WRITE("SCR1",654)(FTYPE(I),I=1,20)
5683        REWIND "SCR1"
5703        READ("SCR1",654)(JTYPE(I),I=1,20)
5723        D0 360 ITYPE=1,3
5743   360  FUREQ(IB,ITYPE)=0.
5763        D0 370 K2=1,KF
5783        IF(JTYPE(K2).E0.JET)IT=1
5803        IF(JTYPE(K2).E0.AVGAS)IT=2
5823        IF(JTYPE(K2).E0.HEL0)IT=3
5843        IF(IT.LT.1)PRINT," FUEL IN PHASE ",J," IS 0F UNKN0WN TYPE"
5863        FUREQ(IB,IT)=FUREQ(IB,IT)+GAREQ(K2)
5883   370  C0NTINUE
5903        D0 372 I=16,21
5923        IK=I-15
5943        J=IFIX(A0M(I)+.005)
5963   372  FUREQ(IB,J)=FUREQ(IB,J)+TENAC(IB,IK)*FLCST(I)
5983        IF(N0DETL.E0.1)G0 T0 375
6003        PRINT 619
6023        PRINT 620,(ATYPE(I),PLREQ(I),I=1,K)
6043   375  D0 380 I=1,K
6063        D0 380 J=1,15
6083        IF(ATYPE(I).NE.IATYPE(J))G0 T0 380
6103        ACREQ(IB,J)=PLREQ(I)
6123        ACFH(IB,J)=HRSREQ(I)
6143   380  C0NTINUE
```

TABLE 23 (Cont)

```
6163        D0 385 I=16,21
6183        J=I-15
6203   385  ACREQ(IB,I)=TENAC(IB,J)
6223        IF(N0DETL.EQ.1)G0 T0 390
6243        PRINT 621
6263        PRINT 622,(FUREQ(IB,I),I=1,3)
6283        G0 T0 400
6303   390  WAG=GAREQ(1)*1.E-6
6323        IF(N0AC.EQ.1)G0 T0 398
6343        PRINT 660,NASNAM(IB),TSTU(IB),SUM3,TNAS(IB),
6363       &T0FF(IB),TENL(IB),TCIV(IB),TBAS(IB),ATYPE(1),PLREQ(1),
6383       &FTYPE(1),WAG
6403        IF(L1.EQ.1)G0 T0 400
6423        D0 395 J1=2,L1
6443        WAG=GAREQ(J1)*1.E-6
6463        IF((K.GE.J1).AND.(KF.GE.J1))PRINT 661,ATYPE(J1),PLREQ(J1),
6483       &FTYPE(J1),WAG
6503        IF((K.LT.J1).AND.(KF.GE.J1))PRINT 662,FTYPE(J1),WAG
6523        IF((K.GE.J1).AND.(KF.LT.J1))PRINT 663,ATYPE(J1),PLREQ(J1)
6543   395  C0NTINUE
6563        G0 T0 400
6583   398  PRINT 660,NASNAM(IB),TSTU(IB),SUM3,TNAS(IB),T0FF(IB),
6603       &TENL(IB),TCIV(IB),TBAS(IB)
6623        N0AC=0
6643   400  C0NTINUE
6663        PRINT 609
6683   410  INPUT,I0P
6703        IF(I0P.EQ.IYES)G0 T0 502
6723        IF(I0P.EQ.N0)CHAIN "PART4*"
6743        PRINT 628
6763        G0 T0 410
6783   500  PRINT 626
6803        IF(ISWTCH(6).EQ.1)ISWTCH(10)=0
6823   501  INPUT 627,I0P
6843        IF(I0P.EQ.IYES)G0 T0 520
6863        IF(I0P.EQ.N0)G0 T0 4
6883        PRINT 628
6903        G0 T0 501
6923   502  ISWTCH(8)=1
6943        G0 T0 4
6963   600  F0RMAT(" PHASE ALL0CATI0N:  ASSIGN EACH PHASE AS--"/1X
6983       &"II,AAAA,.XX"/1X"WHERE: II = PHASE (2 DIGITS); AAAA = BASE"
7003       &" C0DE;"/7X".XX = PERCENT AT BASE (1.0 = 100%)"/1X
```

TABLE 23 (Cont)

```
7023      &"BASE CØDES: CHAS CØRP ELLY"/13X"KING MERI PENS"/13X
7043      &"SAUF WHIT PHAN"/" II = 0 TØ TERMINATE:")
7063    601 FØRMAT(I2,A1,A4,A1,F3.2)
7083    602 FØRMAT(22H BAD FØRMAT--TRY AGAIN)
7103    604 FØRMAT(31H INCØRRECT BASE CØDE--TRY AGAIN)
7123    605 FØRMAT(10H THE VALUE1XF6.2,1X45HGIVEN FØR PERCENT CANNØT EXCE
7143      &ED 1.
7163      &--TRY AGAIN)
7183    606 FØRMAT(5H NEXT)
7203    607 FØRMAT( 7H PHASE I2,49H HAS NØT BEEN ASSIGNED ØR IS ØNLY PART
7223      &LY AS
7243      &SIGNED)
7263    609 FØRMAT(//1X"DØ YØU WISH TØ RE-ALLØCATE PHASES (Y,N)")
7283    619 FØRMAT( //14H AIRCRAFT DATA/1X4HTYPE 4X3HNØ.)
7303    620 FØRMAT(1XA4,2XF5.0)
7323    621 FØRMAT( //10H FUEL DATA/1X4HTYPE2X7HGALLØNS)
7343    622 FØRMAT(1X"JET "1XE9.3/1X"AGAS"1XE9.3/1X"HELØ"1XE9.3)
7363    625 FØRMAT(1X6HPHASE I2,1X46HHAS BEEN ØVER-ASSIGNED, RE-ALLØCATE
7383      &THIS
7403      &HASE )
7423    626 FØRMAT(1X55HDØ YØU WISH TØ KEEP SAME PHASE TØ BASE ASSIGNMENT
7443      & (Y,N))
7463    627 FØRMAT(A1)
7483    628 FØRMAT(1X24HINVALID REPLY--TRY AGAIN )
7503    629 FØRMAT(5X6A4)
7523    630 FØRMAT(5X3E13.6)
7543    650 FØRMAT(A4,5E12.6/4E12.6,I1/6E12.6/6E12.6/E12.6)
7563    651 FØRMAT(5XI3)
7583    652 FØRMAT(5X3A4,I3)
7603    653 FØRMAT(5X5E13.6)
7623    654 FØRMAT(75A1)
7643    655 FØRMAT(A4,6E12.6/6E12.6)
7663    660 FØRMAT(1XA4,F6.0,F7.0,F7.0,3F6.0,F7.0,1XA4,F5.0,1XA4,F7.2)
7683      &F4.0,A4,1X1PE8.3)
7703    661 FØRMAT(51XA4,F5.0,1XA4,F7.2)
7723    662 FØRMAT(61XA4,F7.2)
7743    663 FØRMAT(51XA4,F5.0)
7763    665 FØRMAT(1X"BASE LØADING SUMMARY"/1X"*PERSØNNEL"38X
7783      &3X"*AIRCRAFT *FUEL"/6X"STD.  "12(1H-)"BASE  TØTALS "
7803      &12(1H-)10X"MILLIØN GAL."/1X"NAS  LØAD  PHASE
7823      &NAS   ØFF   ENL   CIV  TØTAL  TYPE  NØ. TYPE AMØUNT")
7843    715 FØRMAT(///1X"NAS--"A4/1X55HPERSØNNEL     STD.LØAD
7863      & ØFFI
```

TABLE 23 (Cont)

```
7883      &CERS ENLISTED CIVILIAN  TØTAL)
7903   716 FØRMAT(1X,3A4,F6.0,F10.0,F9.0,9X,F9.0)
7923   718 FØRMAT(13H ALL PHASES  ,F6.0,F10.0,F9.0,9X,F9.0/)
7943   719 FØRMAT(13H TENANTS     ,6X,F10.0,3F9.0)
7963   720 FØRMAT(13H NAS PERS.   ,6X,F10.0,3F9.0/
7983      &       13H TØTAL BASE  ,6X,F10.0,3F9.0)
8003       END
```

IX. PROGRAM PART4

PROGRAM DESCRIPTION

9.1 PROGRAM PART4 reads the Data File, TABLE*, containing assorted tables for computing facility requirements. In addition, on the first time through, the Assets Position Data File, RPIFI*, and the Cost Data File, INVCO*,are read. Finally, PROGRAM PART4 reads the file RUNWAY containing runway requirements generated within the LSR programs and apportions these requirements to bases by using the phase allocation obtained in PART3.

9.2 Upon entering PART4, the current number of facilities included in the IFRS model is set (i.e., 30). If this is the first time through this program (ISWTCH(10) = 0), the cost file INVCO* is read. Next, the file TABLES* is read.

9.3 Again, if this is the first time through, the Assets Position File (RPIFI*) is read in Subroutine INTRPI. Subroutine RWY1 is called each time through. Then control is transferred to PARTY.

SUBROUTINE RWY1

9.4 In Subroutine RWY1, the user is asked if he wishes to skip the airspace factors and OLF requirements printout. The resulting decision is recorded in the variable ISKIP.

9.5 Next, the runway requirements by phase (file RUNWAY) are read. Aircraft names are checked against a current list and,if no match is found, an error message is displayed. Valid names are recorded according to their order in the stored array IATYPE containing acceptable aircraft names.

9.6 Using the phase to base allocations, the runway requirements are apportioned by base and stored in the array XBR1. Next, airspace and OLF requirements, if requested, are printed by base. If airspace is oversaturated, the user may choose to return to the LSR Generator to constrain outputs from that model, thus reducing airspace saturation. If the choice is made to return to the LSR Generator, control transfers to PART8. If not, printing continues until the airspace and OLF requirements for all bases have been processed. Following this, the user is asked if he wishes to reallocate phases. If so, control transfers to PART3; if not, the computations continue.

9.7 If the user has not transferred out to either PART8 or PART3, the required and available runway tables are sorted in descending order, first on length of runway and second on thickness. When this has been done, the subroutine returns to the main program.

9.8 A flow chart of PROGRAM PART4 is shown in Figure 9. Table 24 contains the variable dictionary of PROGRAM PART4; the program and subroutine dictionary is provided in Table 25. The program listing is shown in Table 26.

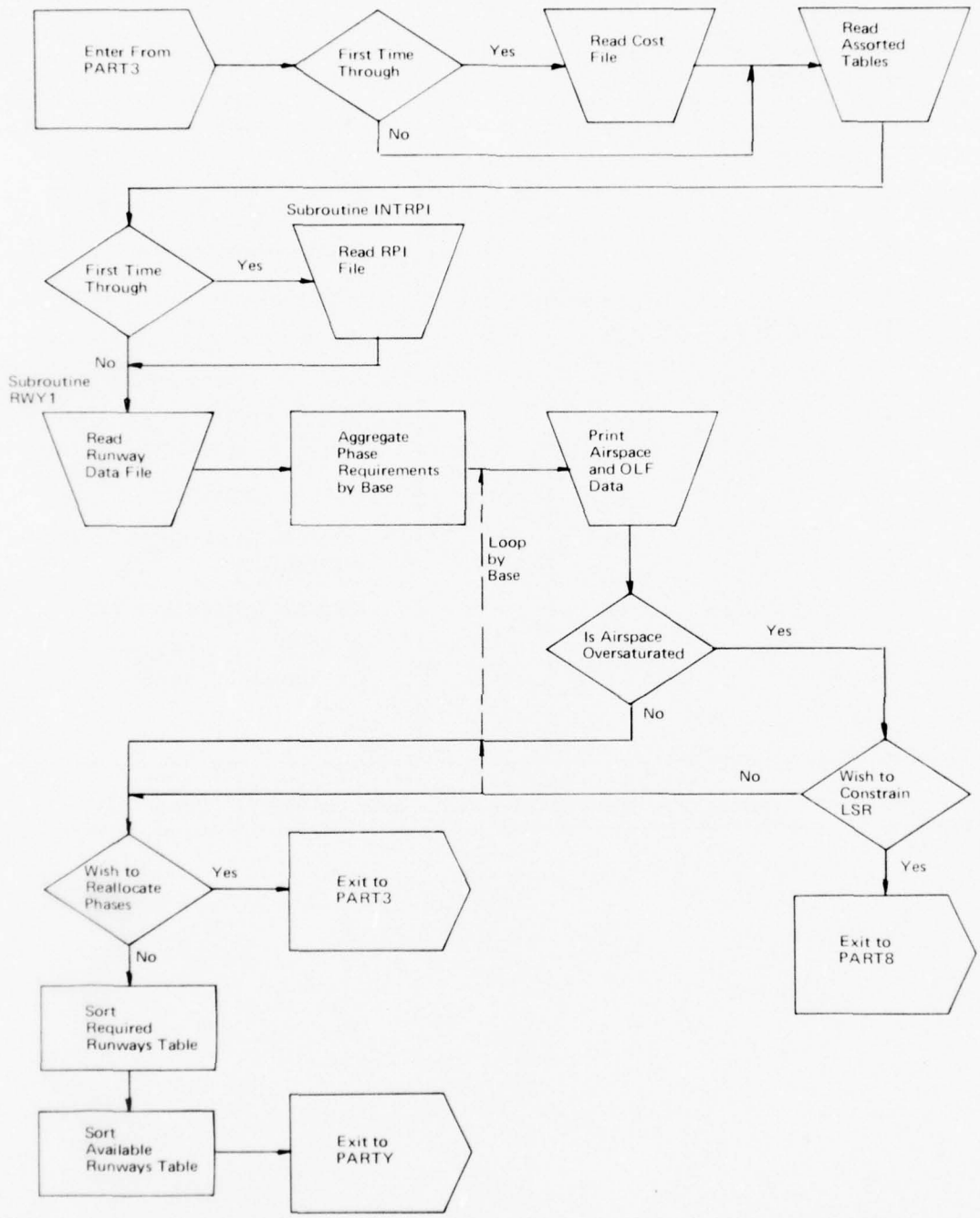


FIGURE 9. PROGRAM PART4 FLOW CHART

TABLE 24

PROGRAM PART4 VARIABLE DICTIONARY*

Location	Variable Name	Dimension	Type	Description
PART4	LINE	1	I	Dummy variable
RWY1	BLNK	1	A	Four blank characters
RWY1	ISKIP	1	A	Print option switch
RWY1	IOP	1	A	Terminal Yes-No response
RWY1	RUNDAT	25,3,3	F	Runway requirements data
RWY1	JACNAM	25,3	A	Aircraft names with runway requirements
RWY1	IFLAG	1	I	Aircraft name match switch
RWY1	ISCR	20	I	Output scratch table
RWY1	TABLE	20,3	F	Air saturation and OLF output table
RWY1	ISPACE	1	I	Oversaturation indicator
RWY1	IACNAM	25,3	I	Aircraft type code for phase I, instruction type I

* For variables in common, see the variable dictionary for PART2, Table 18.

TABLE 25

PROGRAM PART4 PROGRAM AND SUBROUTINE DICTIONARY

PART4	Cost file is read; P-80 tables file is read.
INTRPI	Assets Position Data File is read.
RWY1	Runway requirements and factors file is read and allocated by base.

TABLE 26

PROGRAM PART4 LISTING

```
1004      CØMMØN IYEAR,ISWTCH(10)
1024      CØMMØN ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
1044     &ACFH(9,15),TØFF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
1064     &SØ(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
1084     &IACT(25,3),ACNØ1(25,3),TØFF1(25),EMT1(25)
1104      CØMMØN IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
1124     &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
1144     &RLØAD(21),CØMP(21),FLCST(21),AØM(21),CNAAC(21)
1164      CØMMØN NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
1184     &PRE(9),PØ(9),PS(9),PIE(9),TS(9),TH(9),TNØFF(9),TNENL(9),
1204     &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
1224      CØMMØN FACØST(50,6)
1244      CØMMØN FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
1264     &FAEM(8,2),TANKS(15),TAXITØ(3)
1284      CØMMØN ICØDES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
1304     &XRPI1(9,10,4),XRPI2(3,9)
1324      CØMMØN BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
1344     &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
1364     &NCAT,IYES,NØ,ICØM,GTØTAL,NPH
1384      ALPHA IYES,NØ,ICØM
1404      NCAT=30
1424      IF(ISWTCH(10).NE.0)GØ TØ 8
1444    7 ØPENFILE "INVCØ*"
1464       REWIND "INVCØ*"
1484      DØ 80 I=1,NCAT
1504   80 READ("INVCØ*",603)LINE,(FACØST(I,J),J=1,6)
1524      CLØSEFILE "INVCØ*"
1544    8 ØPENFILE "TABLE*"
1564      REWIND "TABLE*"
1584      READ("TABLE*",604)(FAPW(I1),I1=1,6),((AP(I2,J1),
1604     &I2=1,4),J1=1,3),(GWTAB(J2),J2=1,3),((FAMESS(I3,J3),
1624     &I3=1,7),J3=1,2),((EXCH(I4,J4),I4=1,10),J4=1,2),
1644     &((FAEM(I5,J5),I5=1,8),J5=1,2),(TANKS(I6),I6=1,15),
1664     &(TAXITØ(J6),J6=1,3)
1684      CLØSEFILE "TABLE*"
1704      IF(ISWTCH(10).EQ.0)GØ TØ 10
1724      GØ TØ 20
1744   10 CALL INTRPI
1764   20 CALL RWY1
1784      CHAIN "PARTY*"
1804  603 FØRMAT(V)
1824  604 FØRMAT(6E12.6)
1844      END
```

TABLE 26 (Cont)

a. Subroutine INTRPI

```
1864       SUBROUTINE INTRPI
1884       COMMON IYEAR,ISWTCH(10)
1904       COMMON ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
1924      &ACFH(9,15),TOFF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
1944      &SO(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
1964      &IACT(25,3),ACNO1(25,3),TOFF1(25),EMT1(25)
1984       COMMON IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
2004      &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
2024      &RLOAD(21),COMP(21),FLCST(21),AOM(21),CNAAC(21)
2044       COMMON NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
2064      &PRE(9),PO(9),PS(9),PIE(9),TS(9),TH(9),TNOFF(9),TNENL(9),
2084      &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
2104       COMMON FACOST(50,6)
2124       COMMON FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
2144      &FAEM(8,2),TANKS(15),TAXITO(3)
2164       COMMON ICODES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
2184      &XRPI1(9,10,4),XRPI2(3,9)
2204       COMMON BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
2224      &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
2244      &NCAT,IYES,NO,ICOM,GTOTAL,NPH
2264       ALPHA IYES,NO,ICOM
2284       OPENFILE "RPIFI*"
2304       REWIND "RPIFI*"
2324       DO 30 I=1,NCAT
2344       READ("RPIFI*",600)ICODES(I),(IDES(I,J),J=1,3),IUNITS(I)
2364    30 CONTINUE
2384       DO 40 I=1,9
2404       DO 35 J=1,NCAT
2424    35 READ("RPIFI*",601)LINE,RPI(J,I,1),RPI(J,I,2)
2444       DO 36 J=1,10
2464    36 READ("RPIFI*",601)LINE,(XRPI1(I,J,K),K=1,4)
2484       READ("RPIFI*",601)LINE,(XRPI2(K,I),K=1,3)
2504    40 CONTINUE
2524       CLOSEFILE "RPIFI*"
2544       RETURN
2564   600 FORMAT(4X15,1X3A4,1XA2)
2584   601 FORMAT(V)
2604       END
```

TABLE 26 (Cont)

b. Subroutine RWY1

```
2624       SUBRØUTINE RWY1
2644       CØMMØN IYEAR,ISWTCH(10)
2664       CØMMØN ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
2684      &ACFH(9,15),TØFF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
2704      &SØ(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
2724      &IACT(25,3),ACNØ1(25,3),TØFF1(25),EMT1(25)
2744       CØMMØN IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
2764      &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
2784      &RLØAD(21),CØMP(21),FLCST(21),AØM(21),CNAAC(21)
2804       CØMMØN NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
2824      &PRE(9),PØ(9),PS(9),PIE(9),TS(9),TH(9),TNØFF(9),TNENL(9),
2844      &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
2864       CØMMØN FACØST(50,6)
2884       CØMMØN FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
2904      &FAEM(8,2),TANKS(15),TAXITØ(3)
2924       CØMMØN ICØDES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
2944      &XRPI1(9,10,4),XRPI2(3,9)
2964       CØMMØN BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
2984      &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
3004      &NCAT,IYES,NØ,ICØM,GTØTAL,NPH
3024       DIMENSIØN RUNDAT(25,3,3),JACNAM(25,3),ISCR(20),TABLE(20,3)
3044      &,IACNAM(25,3)
3064       ALPHA IØP,NØ,IYES,BLNK,JACNAM,IATYPE
3084       BLNK="     "
3104       PRINT,"AIRSPACE FACTØRS & ØLF REQUIREMENTS:"
3124       ISKIP=0
3144       PRINT,"SKIP PRINTØUT (Y,N)"
3164     2 INPUT,IØP
3184       IF(IØP.EQ.IYES)GØ TØ 3
3204       IF(IØP.EQ.NØ)GØ TØ 4
3224       PRINT,"BAD REPLY--RETYPE"
3244       GØ TØ 2
3264     3 ISKIP=1
3284     4 ØPENFILE "RUNWAY"
3304       REWIND "RUNWAY"
3324       READ("RUNWAY",600)NPH
3344       READ("RUNWAY",601)((JACNAM(I,J),J=1,3),(RUNDAT(I,J1,1),
3364      &J1=1,3),(RUNDAT(I,J2,2),J2=1,3),(RUNDAT(I,J3,3),J3=1,3),
3384      &I=1,NPH)
3404       CLØSEFILE "RUNWAY"
```

TABLE 26 (Cont)

b. Subroutine RWY1 (Cont)

```
3424C---------REPLACE A/C NAMES BY INTERNAL CØDE
3444       DØ 15 J=1,NPH
3464       DØ 15 K=1,3
3484       IF(JACNAM(J,K).EQ.BLNK)GØ TØ 15
3504       IFLAG=0
3524       DØ 10 I=1,15
3544       IF(JACNAM(J,K).NE.IATYPE(I))GØ TØ 10
3564       IFLAG=1
3584       IACNAM(J,K)=I
3604    10 CØNTINUE
3624       IF(IFLAG.EQ.1)GØ TØ 15
3644       PRINT,"UNRECØGNIZED A/C TYPE IN RUNWAY CØMPUTATIØNS:"
3664       PRINT 602,JACNAM(J,K)
3684    15 CØNTINUE
3704C----------AGGREGATE BY BASE AND A/C TYPE
3724       DØ 100 IB=1,9
3744       IF(NBUSE(IB).EQ.0)GØ TØ 100
3764       K=1
3784       DØ 90 I=1,NPH
3804       C=BPH(IB,I)
3824       IF(C-.01)90,90,18
3844    18 DØ 90 J=1,3
3864       IF(RUNDAT(I,J,1)-.01)90,90,20
3884    20 IF(K-1)25,22,25
3904    22 XBR1(IB,1,1)=RUNDAT(I,J,1)*C
3924       M=IACNAM(I,J)
3944       XBR1(IB,1,2)=RNWYL(M)
3964       XBR1(IB,1,3)=RLØAD(M)+.005
3984       XBR1(IB,1,4)=CØMP(M)+.005
4004       ISCR(1)=M
4024       TABLE(1,1)=RUNDAT(I,J,2)*C
4044       TABLE(1,2)=RUNDAT(I,J,3)*C
4064       TABLE(1,3)=FLØAT(M)+.005
4084       K=2
4104       GØ TØ 90
4124    25 M=IACNAM(I,J)
4144       LM=K-1
4164       DØ 40 L=1,LM
4184       IF(M.NE.ISCR(L))GØ TØ 40
4204       XBR1(IB,L,1)=XBR1(IB,L,1)+RUNDAT(I,J,1)*C
4224       TABLE(L,1)=TABLE(L,1)+RUNDAT(I,J,2)*C
4244       TABLE(L,2)=TABLE(L,2)+RUNDAT(I,J,3)*C
4264       GØ TØ 90
4284    40 CØNTINUE
```

TABLE 26 (Cont)

b. Subroutine RWY1 (Cont)

```
4304        IF(K.EQ.11)GØ TØ 200
4324        XBR1(IB,K,1)=RUNDAT(I,J,1)*C
4344        ISCR(K)=M
4364        XBR1(IB,K,2)=RNWYL(M)
4384        XBR1(IB,K,3)=RLØAD(M)+.005
4404        XBR1(IB,K,4)=CØMP(M)+.005
4424        TABLE(K,1)=RUNDAT(I,J,2)*C
4444        TABLE(K,2)=RUNDAT(I,J,3)*C
4464        TABLE(K,3)=FLØAT(M)+.005
4484        K=K+1
4504    90  CØNTINUE
4524C---------PRINT AIRSPACE AND ØLF DATA
4544        IF(ISKIP.EQ.1)GØ TØ 100
4564        PRINT 603,NASNAM(IB)
4584        KM=K-1
4604        IF(KM.EQ.0)GØ TØ 98
4624        ISPACE=0
4644        DØ 95 I=1,KM
4664        M=IFIX(TABLE(I,3))
4684        IF(TABLE(I,1).GT.1.)ISPACE=1
4704    95  PRINT 604,IATYPE(M),TABLE(I,1),TABLE(I,2)
4724        IF(ISPACE.NE.1)GØ TØ 100
4744        PRINT,"NØTE: AIRSPACE IS ØVER-SATURATED"
4764        PRINT,"DØ YØU WISH TØ CØNSTRAIN LSR ØUTPUT (Y,N)"
4784    96  INPUT,IØP
4804        IF(IØP.EQ.NØ)GØ TØ 100
4824        IF(IØP.EQ.IYES)CHAIN "PART8*"
4844        PRINT,"INVALID REPLY--RETYPE"
4864        GØ TØ 96
4884    98  PRINT,"NØ REQUIREMENTS"
4904   100  CØNTINUE
4924        IF(ISKIP.EQ.1)GØ TØ 120
4944        PRINT,"DØ YØU WISH TØ RE-ALLØCATE PHASES TØ BASES (Y,N)"
4964   105  INPUT,IØP
4984        IF(IØP.EQ.NØ)GØ TØ 120
5004        IF(IØP.EQ.IYES)GØ TØ 110
5024        PRINT,"INVALID REPLY--RETYPE"
5044        GØ TØ 105
5064   110  ISWTCH(8)=1
5084        CHAIN "PART3*"
```

TABLE 26 (Cont)

b. Subroutine RWY1 (Cont)

```
5104C---------SØRT REQUIREMENTS FIRST BY LENGTH, THEN THICKNESS
5124  120 DØ 180 IB=1,9
5144      IF(NBUSE(IB).EQ.0)GØ TØ 180
5164      DØ 130 I=1,9
5184      JL=I+1
5204      DØ 130 J=JL,10
5224      IF(XBR1(IB,I,2).GT.XBR1(IB,J,2))GØ TØ 130
5244      DØ 124 J1=1,4
5264      TEMP=XBR1(IB,I,J1)
5284      XBR1(IB,I,J1)=XBR1(IB,J,J1)
5304  124 XBR1(IB,J,J1)=TEMP
5324  130 CØNTINUE
5344      DØ 140 I=1,9
5364      JL=I+1
5384      DØ 135 J=JL,10
5404      IF((XBR1(IB,I,2)-XBR1(IB,J,2)).GT.10.)GØ TØ 140
5424      IF(XBR1(IB,I,3).GT.XBR1(IB,J,3))GØ TØ 135
5444      DØ 132 J1=1,4
5464      TEMP=XBR1(IB,I,J1)
5484      XBR1(IB,I,J1)=XBR1(IB,J,J1)
5504  132 XBR1(IB,J,J1)=TEMP
5524  135 CØNTINUE
5544  140 CØNTINUE
5564C---------SØRT AVAILABLE
5584      DØ 150 I=1,9
5604      JL=I+1
5624      DØ 150 J=JL,10
5644      IF(XRPI1(IB,I,2).GT.XRPI1(IB,J,2))GØ TØ 150
5664      DØ 144 J1=1,4
5684      TEMP=XRPI1(IB,I,J1)
5704      XRPI1(IB,I,J1)=XRPI1(IB,J,J1)
5724  144 XRPI1(IB,J,J1)=TEMP
5744  150 CØNTINUE
5764      DØ 160 I=1,9
5784      JL=I+1
5804      DØ 155 J=JL,10
5824      IF((XRPI1(IB,I,2)-XRPI1(IB,J,2)).GT.10.)GØ TØ 160
5844      IF(XRPI1(IB,I,3).GT.XRPI1(IB,J,3))GØ TØ 155
5864      DØ 152 J1=1,4
5884      TEMP=XRPI1(IB,I,J1)
5904      XRPI1(IB,I,J1)=XRPI1(IB,J,J1)
5924  152 XRPI1(IB,J,J1)=TEMP
```

TABLE 26 (Cont)

b. Subroutine RWY1 (Cont)

```
5944   155 CØNTINUE
5964   160 CØNTINUE
5984   180 CØNTINUE
6004       RETURN
6024   200 PRINT,"RUNWAY REQUIREMENTS EXCEED AVAILABLE TABLE SPACE"
6044       PRINT,"PRØGRAM ABØRT"
6064       STØP
6084   600 FØRMAT(5XI3)
6104   601 FØRMAT(5X3A4/5X3E13.6/5X3E13.6/5X3E13.6)
6124   602 FØRMAT(1XA4)
6144   603 FØRMAT(1X"NAS--"A4/3X"TYPE A/C"4X"AIRSPACE"3X"ØLF'S"/
6164      &16X"FACTØR"2X"REQUIRED")
6184   604 FØRMAT(5XA4,7XF5.2,4XF5.2)
6204       END
```

X. PROGRAM PARTY

PROGRAM DESCRIPTION

10.1 PROGRAM PARTY [1] computes runway deficiencies, if any, records their cost, and computes resulting deficiencies in taxiways and runway lighting.

10.2 Upon entering PARTY, a tolerance on runway length is set (i.e., 10 ft) and two alpha variables are initialized for later printouts. The user is asked if he wants to skip the printouts and the resulting decision is stored in ISKIP. Then GTOTAL, the total systems cost, is initialized to zero.

10.3 The program first checks to see which bases are in use. For those in use, two tables which will contain taxiway requirements and runway lighting requirements are initialized to zero. Next, the runways available and required are printed if the print option is selected and the runway table pointers (i.e., internal indices), I,J, are initialized to 1.

10.4 The next portion of the program compares the length and thickness of the J^{th} required runway with the length and thickness of the I^{th} available runway. When the length of the I^{th} available runway is less than the tolerance value, i.e., a value less than the tolerance of 10 ft, no more runways are available at the base under consideration and the program branches to the "build new" computations.

10.5 If the length requirement J is less than 500 ft and the available runway length I is greater than 500 ft, the aircraft is a helicopter and it is assumed that there are no runway deficits.

10.6 If the length and thickness required are less than or equal to available, the program branches to the amount comparison computations. If requirements exceed

---

[1] This program constitutes part of the Runway Submodel discussed in Appendix H of Volume II.

available assets, the existing runway considered is upgraded to meet the requirements and the resulting taxiways and runway lighting requirements are computed. Finally, the upgrading cost is computed and the construction information is placed in TABLE.

10.7 The next portion of the program is the amount-comparison computation. [2/] If the available runway considered (as determined by the pointer I) does not meet the considered amount required (as determined by the pointer J), the pointer for the available table is advanced and the program transfers back to compare length and thickness requirements against available assets. If the requirement is less than the available amount, the excess is recorded, the requirements table pointer is advanced, and the excess is applied to the next requirement.

10.8 At any point in which new runways are required, i.e., when available runways are exhausted and requirements remain, the program tests ITEST to see whether it is positive or negative. As each new runway is built, the sign changes. ITEST positive or negative indicates primary or crosswind runway construction, respectively. The resulting taxiway and runway lighting requirements are then computed. Finally, the construction cost is computed.

10.9 When all requirements have been met if there are no deficiencies, then the computations proceed to the next base after a "no runway deficits" printout prints under the print option. Under the print option, the deficits (if any) are printed in tabular form and the user has the option to delete any construction requirements made in the program. He also has the choice at this point to return to constrain LSR output. A Yes response transfers control within the program to PART8; a No response causes the computations to continue. The runway investment costs are then summed and accumulated into the total systems cost.

10.10 Once all bases have been processed, the total runway investment cost is printed and control transfers to PART5.

10.11 A flow chart of PROGRAM PARTY is shown in Figure 10. Table 27 contains the variable dictionary of PROGRAM PARTY; the program dictionary is provided in Table 28. The program listing is shown in Table 29.

[2/] The number of pure runways required.

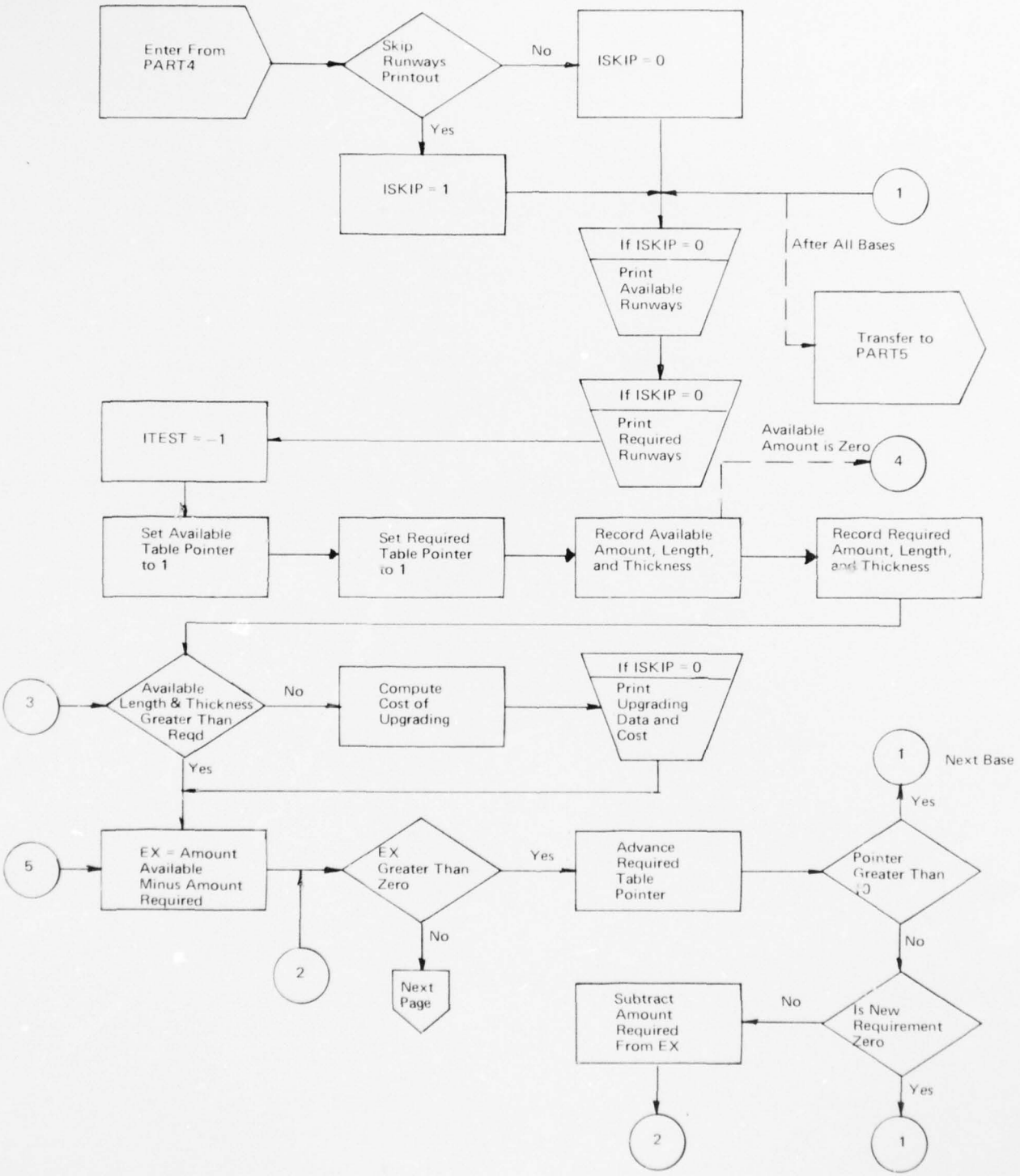


FIGURE 10. PROGRAM PARTY FLOW CHART

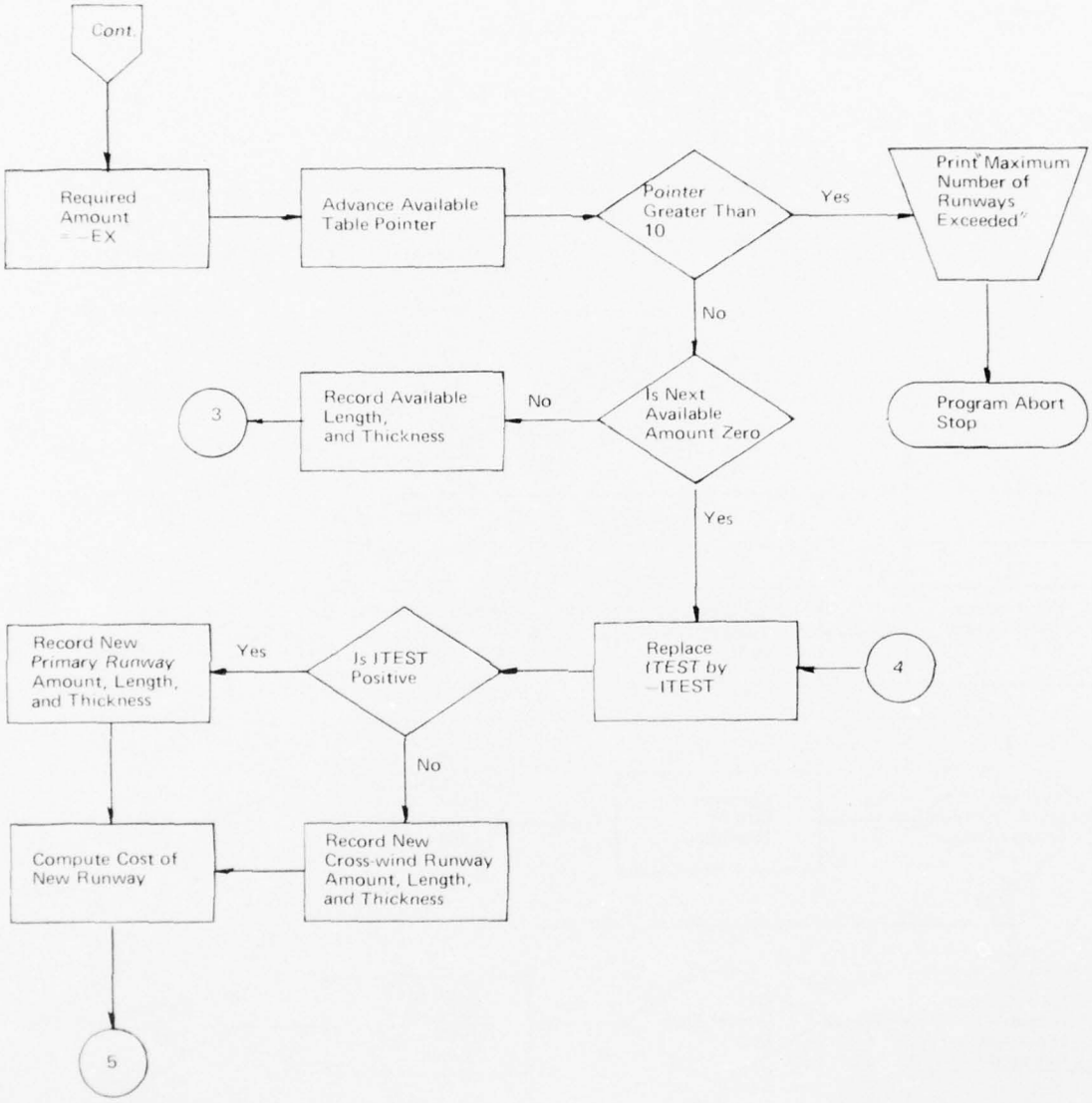


FIGURE 10 (Cont)

FIGURE 9. PROGRAM PART4 FLOW CHART

TABLE 27

PROGRAM PARTY VARIABLE DICTIONARY *

Location	Variable Name	Dimension	Type	Description
PARTY	TOL	1	F	Runway length tolerance, ft
PARTY	TYPE	2	A	Runway types
PARTY	ISKIP	1	I	Print option switch
PARTY	IOP	1	A	Terminal Yes-No response
PARTY	STABLE	10	F	Taxiway requirements table
PARTY	RTABLE	10,2	F	Runway requirements table
PARTY	ITEST	1	I	Primary-crosswind runway switch
PARTY	KTABLE	1	I	Table index
PARTY	A	1	F	Amount of available runway
PARTY	R	1	F	Amount of required runway
PARTY	LR	1	F	Required runway length
PARTY	LA	1	F	Available runway length
PARTY	TR	1	F	Thickness factor for required runway
PARTY	TA	1	F	Thickness factor for available runway
PARTY	AL	1	F	Runway length actual
PARTY	MT	1	I	Thickness factor actual
PARTY	DL	1	F	Additional runway length
PARTY	COCOST	9,12	F	Cost of 1 sq yd of runway composition J, thickness I
PARTY	THCOST	2,9	F	Thickness factor for costing runway of composition I, thickness J

* For common variables, see the variable dictionary for PART2, Table 18.

TABLE 27 (Cont)

Location	Variable Name	Dimension	Type	Description
PARTY	EX	1	F	Excess runway available
PARTY	IWR	1	I	Runway type indicator
PARTY	COST	1	F	Runway construction cost

TABLE 28

PROGRAM PARTY PROGRAM DICTIONARY

PARTY	Computes runway, taxiway, and runway lighting deficiencies

TABLE 29

PROGRAM PARTY LISTING

```
10011       COMMON IYEAR,ISWTCH(10)
10031       COMMON ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
10051      &ACFH(9,15),TOFF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
10071      &SO(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
10091      &IACT(25,3),ACNO1(25,3),TOFF1(25),EMT1(25)
10111       COMMON IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
10131      &AHM(21),ACM(21),ASM1(21),ASM2(21),B(21,3),RNWYL(21),
10151      &RLOAD(21),COMP(21),FLCST(21),AOM(21),CNAAC(21)
10171       COMMON NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
10191      &PRE(9),PO(9),PS(9),PIE(9),TS(9),TH(9),TNOFF(9),TNENL(9),
10211      &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
10231       COMMON FACOST(50,6)
10251       COMMON FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
10271      &FAEM(8,2),TANKS(15),TAXITO(3)
10291       COMMON ICODES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
10311      &XRPI1(9,10,4),XRPI2(3,9)
10331       COMMON BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
10351      &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
10371      &NCAT,IYES,NO,ICOM,GTOTAL,NPH
10391       DIMENSION TABLE(10,4),TYPE(2),THCOST(2,9),STABLE(10)
10411      &,COCOST(9,2),RTABLE(10,2)
10431       REAL LR,LA
10451       ALPHA TYPE,IOP,NO,IYES
10471       DATA THCOST/6.,11.,6.,14.,7.,16.,8.,18.,9.,19.,10.,23.,
10491      &10.,24.,10.,28.,12.,35./
10511       DATA COCOST/10.84,10.84,11.83,12.06,13.89,13.89,14.33,15.66,
10531      &16.95,4.65,5.53,6.40,7.28,7.98,8.27,8.57,10.32,12.40/
10551       TOL=10.
10571       TYPE(1)="MAIN"
10591       TYPE(2)="XWND"
10611       PRINT,"DO YOU WANT TO SKIP RUNWAY REQUIREMENTS OUTPUT (Y,N)"
10631       ISKIP=0
10651     5 INPUT,IOP
10671       IF(IOP.EQ.IYES)GO TO 7
10691       IF(IOP.EQ.NO)GO TO 8
10711       PRINT,"INVALID REPLY--RETYPE"
10731       GO TO 5
10751     7 ISKIP=1
10771     8 IF(ISKIP.EQ.0)PRINT,"RUNWAY REQUIREMENTS BY BASE:"
10791       GTOTAL=0.
```

TABLE 29 (Cont)

```
10811       DØ 500 IB=1,9
10831       IF(NBUSE(IB).EQ.0)GØ TØ 500
10851       XDEF2(IB)=0.
10871       IF(ISKIP.EQ.0)PRINT 600,NASNAM(IB)
10891       DØ 10 I=1,10
10911       STABLE(I)=0.
10931       RTABLE(I,1)=0.
10951       RTABLE(I,2)=0.
10971       IF (XRPI1(IB,I,1).LE..005)GØ TØ 20
10991       M=IFIX(XRPI1(IB,I,3)+.005)
11011    10 IF(ISKIP.EQ.0)PRINT 601,(XRPI1(IB,I,J),J=1,2),M
11031    20 IF(ISKIP.EQ.0)PRINT 602
11051       DØ 30 I=1,10
11071       IF(XBR1(IB,I,1).LE..005)GØ TØ 40
11091       M=IFIX(XBR1(IB,I,3)+.005)
11111    30 IF(ISKIP.EQ.0)PRINT 601,(XBR1(IB,I,J),J=1,2),M
11131    40 ITEST=-1
11151       KTABLE=1
11171       I=1
11191       J=1
11211       A=XRPI1(IB,I,1)
11231       R=XBR1(IB,J,1)
11251       LR=XBR1(IB,J,2)
11271       TR=XBR1(IB,J,3)-.005
11291    50 LA=XRPI1(IB,I,2)
11311       IF(LA-TØL)130,55,55
11331    55 TA=XRPI1(IB,I,3)+.01
11351       IF((LR.LT.500.).AND.(LA.GE.500.))GØ TØ 400
11371    60 IF((LR.LE.(LA+TØL)).AND.(TR.LE.TA))GØ TØ 90
11391       MA=IFIX(TA)
11411       MR=IFIX(TR+.005)
11431       AL=AMAX1(LR,LA)
11451       MT=MAXØ(MA,MR)
11471       IF(ISKIP.EQ.0)PRINT 603,LA,AL,MA,MT
11491       K2=IFIX(XBR1(IB,J,4)+.005)
11511       K1=IFIX(XRPI1(IB,I,4)+.005)
11531       DL=LR-LA
11551       IF(DL.LT.TØL)GØ TØ 65
11571       NS1=1
11591       IF(LR.GT.TAXITØ(1))NS1=2
11611       IF(LR.GT.TAXITØ(2))NS1=3
11631       NS2=1
11651       IF(LA.GT.TAXITØ(1))NS2=2
11671       IF(LA.GT.TAXITØ(2))NS2=3
11691       NS=NS1-NS2
```

TABLE 29 (Cont)

```
11711       DN=0.
11731       IF(NS.GE.1)DN=FLØAT(NS)
11751       STABLE(KTABLE)=DN*27187.+54375.
11771       IF(I.EQ.1)STABLE(KTABLE)=STABLE(KTABLE)+75.*DL
11791       STABLE(KTABLE)=STABLE(KTABLE)/9.
11811    65 GØ TØ (70,80),K1
11831    70 CØST=(DL*CØCØST(MT,1)+.84*(THCØST(1,MT)-THCØST(1,MA))*
11851      &LA)*(2./90.)*.95
11871       GØ TØ 85
11891    80 CØST=(LA*(.70+.65*(THCØST(2,MT)-THCØST(2,MA)))+
11911      &DL*CØCØST(MT,2))*(2./90.)*.95
11931       RTABLE(KTABLE,2)=RTABLE(KTABLE,2)+1.
11951    85 IF(ISKIP.EQ.0)PRINT 604,CØST
11971       TABLE(KTABLE,1)=FLØAT(I)+.005
11991       TABLE(KTABLE,2)=LR
12011       TABLE(KTABLE,3)=FLØAT(MT)+.005
12031       TABLE(KTABLE,4)=CØST
12051       RTABLE(KTABLE,1)=RTABLE(KTABLE,1)+DL
12071       KTABLE=KTABLE+1
12091    90 EX=A-R
12111   100 IF(EX.LT.0)GØ TØ 120
12131       J=J+1
12151       IF(J.GE.11)GØ TØ 400
12171       R=XBR1(IB,J,1)
12191       IF(R-.01) 400,400,110
12211   110 EX=EX-R
12231       LR=XBR1(IB,J,2)
12251       TR=XBR1(IB,J,3)
12271       IF(LR.LT.500.)GØ TØ 400
12291       GØ TØ 100
12311   120 R=-EX
12331       I=I+1
12351       IF(I.GE.11)GØ TØ 550
12371       A=XRPI1(IB,I,1)
12391       IF(A-.01) 130,130,50
12411   130 ITEST=-ITEST
12431       IWR=1
12451       IF(ITEST.GT.0)GØ TØ 140
12471       IWR=2
```

TABLE 29 (Cont)

```
12491   140 MT=IFIX(TR+.005)
12511       IF(ISKIP.EQ.0)PRINT 605,TYPE(IWR),XBR1(IB,J,2),M
12531       A=WR(IB,IWR)
12551       LA=LR
12571       TA=TR
12591       K1=IFIX(XBR1(IB,J,4)+.005)
12611       RTABLE(KTABLE,1)=RTABLE(KTABLE,1)+LA
12631       RTABLE(KTABLE,2)=RTABLE(KTABLE,2)+2.
12651       IF(K1.EQ.1)GØ TØ 150
12671       CØST=LR*CØCØST(MT,2)*(2./90.)*.95
12691       GØ TØ 160
12711   150 CØST=LR*CØCØST(MT,1)*(2./90.)*.95
12731   160 NS1=1
12751       IF(LR.GT.TAXITØ(1))NS1=2
12771       IF(LR.GT.TAXITØ(2))NS1=3
12791       DN=FLØAT(NS1)
12811       STABLE(KTABLE)=37500.*DN+186250.
12831       IF(I.EQ.1)STABLE(KTABLE)=STABLE(KTABLE)+75.*LR
12851       STABLE(KTABLE)=STABLE(KTABLE)/9.
12871       GØ TØ 85
12891C---------DETERMINE IF DEFICITS WILL BE MADE UP
12911   400 IF(KTABLE.EQ.1)GØ TØ 450
12931       KHI=KTABLE-1
12951       IF(ISKIP.EQ.1)GØ TØ 440
12971       PRINT,"SUMMARY ØF RUNWAY UPGRADE/CØNSTRUCTIØN"
12991       PRINT 606
13011       DØ 410 K=1, KHI
13031       IVAL=IFIX(TABLE(K,3)+.005)
13051   410 PRINT 607,K,TABLE(K,2),IVAL,TABLE(K,4)
13071       PRINT," WILL THESE DEFICITS BE MADE UP (Y,N)"
13091   415 INPUT, IØP
13111       IF(IØP.EQ.IYES) GØ TØ 440
13131       IF (IØP.EQ.NØ) GØ TØ 420
13151       PRINT," INVALID REPLY--RETYPE"
13171       GØ TØ 415
13191   420 PRINT,"DØ YØU WISH TØ CØNSTRAIN LSR ØUTPUT (Y,N)"
13211   421 INPUT,IØP
13231       IF(IØP.EQ.IYES)CHAIN "PART8*"
13251       IF(IØP.EQ.NØ)GØ TØ 422
13271       PRINT,"INVALID REPLY--RETYPE"
13291       GØ TØ 421
```

TABLE 29 (Cont)

```
13311   422 PRINT," WHICH RUNWAYS WØN'T BE BUILT ØR UPGRADED"
13331       PRINT," ENTER A NUMBER FRØM PRECEDING TABLE--"
13351       PRINT," TYPE 0 (ZERØ) WHEN FINISHED"
13371   425 INPUT,IVAL
13391       IF (IVAL.EQ.0) GØ TØ 440
13411       TABLE(IVAL,4)=0.
13431   430 PRINT," NEXT--"
13451       GØ TØ 425
13471   440 DØ 445 K=1,KHI
13491       IF(TABLE(K,4)-.005)445,445,442
13511   442 GTØTAL=GTØTAL+TABLE(K,4)
13531       XDEF2(IB)=XDEF2(IB)+STABLE(K)
13551       XDEF3(1,IB)=XDEF3(1,IB)+RTABLE(K,1)
13571       XDEF3(2,IB)=XDEF3(2,IB)+RTABLE(K,2)
13591       IVAL=IFIX(TABLE(K,1))
13611       IF (XRPI1(IB,IVAL,2).LT.TABLE(K,2))
13631      &XRPI1(IB,IVAL,2)=TABLE(K,2)
13651       IF(XRPI1(IB,IVAL,3).LT.TABLE(K,3))XRPI1(IB,IVAL,3)
13671      &=TABLE(K,3)
13691   445 CØNTINUE
13711       GØ TØ 500
13731   450 IF(ISKIP.EQ.0)PRINT," NØ RUNWAY DEFICITS"
13751   500 CØNTINUE
13771       PRINT,"

                  TØTAL RUNWAY INVESTMENT FØR CURRENT YEAR (THØUS.):
13791       PRINT 608,GTØTAL
13811       CHAIN "PART5*"
13831   550 PRINT," MAX NØ. ØF RUNWAY TYPES EXCEEDED"
13851       PRINT," PRØGRAM ABØRT"
13871       STØP
13891   600 FØRMAT(//1X"NAS--"A4/1X"AVAILABLE:"/
13911      &1X"AMØUNT LENGTH THICKNESS")
13931   601 FØRMAT(1XF5.2,2XF6.0,4XI2)
13951   602 FØRMAT(1X"REQUIRED:"/1X"AMØUNT LENGTH THICKNESS")
13971   603 FØRMAT(1X"UPGRADE: LENGTH:"F6.0,1X"TØ "F6.0/
13991      &1X"THICKNESS: "I6,1X"TØ "I6)
14011   604 FØRMAT(1X"CØST:"F8.0," (THØUS.)")
14031   605 FØRMAT(1X"BUILD NEW "A4"; LENGTH: "F6.0," THICKNESS: "I2)
14051   606 FØRMAT(1X"NØ. LENGTH THICKNESS CØST (THØUS.)")
14071   607 FØRMAT(1XI2,2XF6.0,4XI2,4XF6.0)
14091   608 FØRMAT(15XF7.0)
14111       END
```

XI. PROGRAM PART5

PROGRAM DESCRIPTION

11.1 PROGRAM PART5[1/] computes facilities requirements based on base loading data and the phase allocation established in PART3. In addition, the level of print detail for the remaining programs is recorded.

11.2 Upon entering PART5, the facilities requirements table is set to zero and the user is asked to supply the level of print detail, which is stored in ISWTCH(9). Then, for all bases in use, each facility requirement is computed, one after another, using the equations described in the annex to Appendix D, Volume II. After all the computations are made, control transfers to PARTX.

11.3 A flow chart of PROGRAM PART5 is shown in Figure 11. Table 30 contains the variable dictionary of PROGRAM PART5; the program dictionary is provided in Table 31. The program listing is shown in Table 32.

1/ This program is essentially the Facilities Requirements Submodel discussed in Volumes I and II.

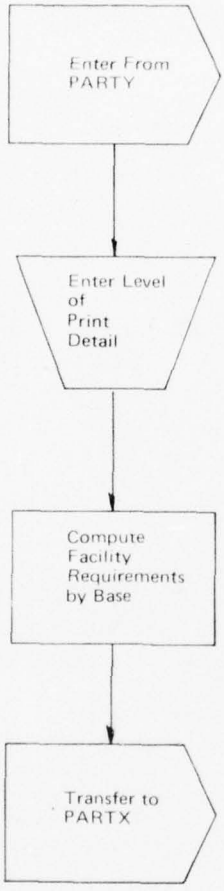


FIGURE 11. PROGRAM PART5 FLOW CHART

TABLE 30

PROGRAM PART5 VARIABLE DICTIONARY *

Location	Variable Name	Dimension	Type	Description
PART5	TW	9	F	Parking apron taxiwidth at base I
PART5	ICHOICE	1	I	Print level input from terminal
PART5	PHASP	1	F	Phase and tenant personnel
PART5	NTRAP	1	I	First pass test switch
PART5	IP	1	I	Integral number of parked aircraft per column
PART5	T	1	F	Residual column length in parking apron
PART5	C	1	F	Number of columns of parking apron required for a type aircraft
PART5	IS	1	1	Variable C rounded to next higher integer
PART5	V	1	F	Parking apron width uncorrected
PART5	IV	1	I	Corrected parking apron width, multiple of 75 ft
PART5	CON1	1	F	Temporary storage
PART5	QH	1	F	Hangar modules required
PART5	QC	1	F	Crew and equipment modules required
PART5	QS1	1	F	Basic shop modules required
PART5	QS2	1	F	Supplementary shop modules required
PART5	TK	1	F	QH rounded up to next integer
PART5	TC	1	F	QC rounded up to next integer
PART5	TS1	1	F	QS1 rounded up to next integer
PART5	TS2	1	F	QS2 rounded up to next integer
PART5	PWP	1	F	Public works personnel

* For variables in common, see the variable dictionary for PART2, Table 18.

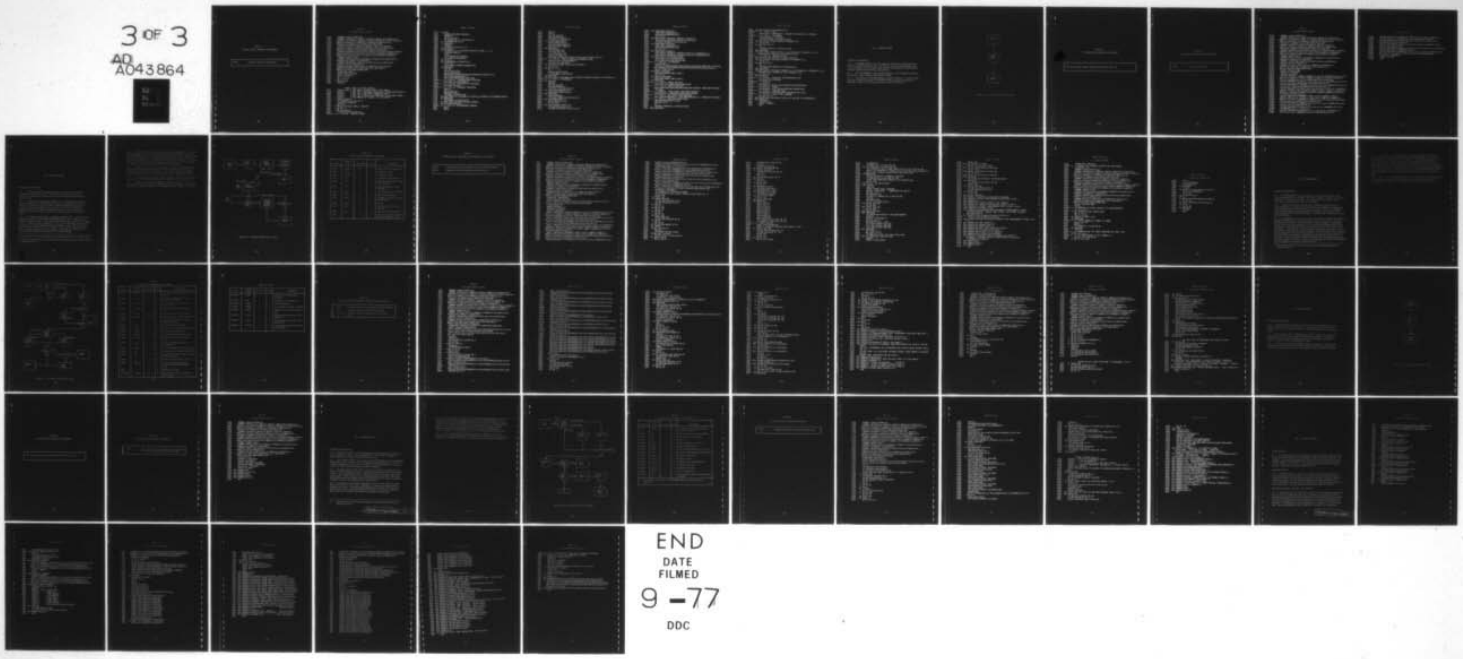
TABLE 30 (Cont)

Location	Variable Name	Dimension	Type	Description
PART5	WSA1	1	F	Warehouse space required for aircraft
PART5	WSA2	1	F	Shed space required for aircraft
PART5	WSP1	1	F	Warehouse space required for personnel
PART5	WSP2	1	F	Shed space required for personnel
PART5	FAEMC	8,2	F	Equivalent to FAEM**

** See Table 18.

AD-A043 864 OPERATIONS RESEARCH INC SILVER SPRING MD F/G 5/1
DEVELOPMENT OF A PRELIMINARY AUTOMATED TOTAL SYSTEMS MODEL FOR --ETC(U)
FEB 70 T N KYLE, R D HEILBRON, J D AVILA N00025-67-C-0031
UNCLASSIFIED ORI-TR-583-VOL-4 NL

3 OF 3
AD A043864



END
DATE FILMED
9 -77
DDC

TABLE 31

PROGRAM PART5 PROGRAM DICTIONARY

PART5	Computes facilities requirements

TABLE 32

PROGRAM PART5 LISTING

```
1005       C0MM0N IYEAR,ISWTCH(10)
1015       C0MM0N ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
1025      &ACFH(9,15),T0FF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
1035      &S0(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
1045      &IACT(25,3),ACN01(25,3),T0FF1(25),EMT1(25)
1055       C0MM0N IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
1065      &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
1075      &RL0AD(21),C0MP(21),FLCST(21),A0M(21),CNAAC(21)
1085       C0MM0N NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
1095      &PRE(9),P0(9),PS(9),PIE(9),TS(9),TH(9),TN0FF(9),TNENL(9),
1105      &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
1115       C0MM0N FAC0ST(50,6)
1125       C0MM0N FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
1135      &FAEMC(8,2),TANKS(15),TAXIT0(3)
1145       C0MM0N IC0DES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
1155      &XRPI1(9,10,4),XRPI2(3,9)
1165       C0MM0N BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
1175      &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
1185      &NCAT,IYES,N0,IC0M,GT0TAL,NPH
1195       DIMENSI0N TW(9)
1205       ALPHA IYES,N0,IC0M
1215       DATA TW/9*150./
1225       D0 5 I=1,50
1235       D0 5 J=1,9
1245     5 BR(I,J)=0.
1255       PRINT,"

                 WHICH LEVEL 0F PRINT DETAIL"
1265       PRINT,"TYPE 1 F0R T0TAL SYSTEMS C0ST (TSC) 0NLY"
1275       PRINT,"      2 F0R TSC & DETAILED FACILITIES EXCESS-DEFICIENCY"
1285       PRINT,"      3 F0R TSC & NAS C0ST SUMMARIES 0NLY"
1295       PRINT,"      4 F0R TSC & FACILITIES DEFICITS & NAS C0ST SUMM."
1305       PRINT,"      5 F0R CH0ICE 0F DETAILS (IF DESIRED)"
1315     7 INPUT,ICH0ICE
1325       IF(ICH0ICE.LT.1)G0 T0 8
1335       ISWTCH(9)=ICH0ICE
1345       G0 T0 9
1355     8 PRINT,"INVALID REPLY--RETYPE"
1365       G0 T0 7
1375     9 D0 500 NB=1,9
1385       IF(NBUSE(NB))10,500,10
1395C---------113-20  PARKING APR0N
```

TABLE 32 (Cont)

```
1405    10 V=0.
1415       PHASP=TBAS(NB)-TNAS(NB)
1425       NTRAP=0
1435       DØ 30 J=1,21
1445       IF(ACREQ(NB,J)-.01)30,30,11
1455    11 IF(NTRAP)15,12,15
1465    12 GO=ACD(J)
1475       NTRAP=1
1485    15 IF(ACD(J)-GO)16,17,17
1495    16 GO=ACD(J)
1505    17 IF(ACC(J).LT..01)PRINT,"ACC(J)=0 FØR J = ",J
1515       P=AD(NB)/ACC(J)
1525       IP=P
1535       T=(P-FLØAT(IP))*ACC(J)
1545       IF(T-ACB(J))25,20,20
1555    20 IP=IP+1
1565    25 C=ACREQ(NB,J)/FLØAT(IP)
1575       IS=C+.99
1585       V=V+(ACA(J)+ACD(J))*FLØAT(IS)
1595    30 CØNTINUE
1605       V=V-GO
1615       IV=(V/75.)+.99
1625       AW=75.*FLØAT(IV)
1635       BR(28,NB)=(AW+2.*TW(NB))*(AD(NB)+2.*TW(NB))/9.
1645       BR(1,NB)=AW*AD(NB)/9.
1655       BR(27,NB)=BR(28,NB)-BR(1,NB)
1665C---------125-40  DISTRIBUTIØN PIPELINE
1675       BR(2,NB)=.001375*PHASP
1685C---------141-40  AIRCRAFT ØP BUILDING
1695       BR(3,NB)=16956.
1705C---------171-10  ACADEMIC BUILDING
1715       CØN1=0.
1725       DØ 50 J=1,25
1735       CØN2=BPH(NB,J)
1745       IF(CØN2-.01)50,50,40
1755    40 CØN1=CØN1+SI(J)*(ASH(J,1)+ASH(J,2)+ASH(J,3))*(CØN2/CU(NB))
1765    50 CØNTINUE
1775       IF(CU(NB)-.01)60,60,55
1785    55 CØN1=CØN1+TS(NB)*(TH(NB)/CU(NB))
1795    60 BR(4,NB)=CØN1*75.
1805C---------211-10  MAINTENANCE HANGAR
1815       QH=0.
1825       QC=0.
```

TABLE 32 (Cont)

```
1835        QS1=0.
1845        QS2=0.
1855        D0 100 J=1,20
1865        C0N1=ACREQ(NB,J)
1875        IF(C0N1-.01)80,80,70
1885     70 QH=QH+C0N1/AHM(J)
1895        QC=QC+C0N1/ACM(J)
1905        QS1=QS1+C0N1/ASM1(J)
1915        QS2=QS2+C0N1/ASM2(J)
1925     80 C0NTINUE
1935        TK=IFIX(QH+.99)
1945        TC=IFIX(QC+.99)
1955        TS1=IFIX(QS1+.99)
1965        TS2=IFIX(QS2+.99)
1975        BR(5,NB)=13698.*TK+8450.*(TS1+TS2)+10400.*(TC+.5)
1985C---------219-10  PUBLIC W0RKS MAINTENANCE SH0P
1995        PWP=75.77673+.04459*PHASP
2005        IF(PWP-100.)82,82,85
2015     82 BR(6,NB)=((PWP-100.)/100.)*FAPW(1)
2025        G0 T0 110
2035     85 D0 90 I=2,6
2045        K=I
2055        C0N1=FL0AT(K)*100.
2065        IF(PWP-C0N1)100,100,90
2075     90 C0NTINUE
2085    100 BR(6,NB)=((PWP-C0N1+100.)/100.)*(FAPW(K)-FAPW(K-1))+FAPW(K-1)
2095C---------442-10  WAREH0USE
2105    110 WSA1=0.
2115        WSA2=0.
2125        D0 120 J=1,20
2135        C0N1=ACREQ(NB,J)
2145        IF(C0N1-.01)120,120,115
2155    115 WSA1=WSA1+C0N1*A(J,1)
2165        WSA2=WSA2+C0N1*A(J,2)
2175    120 C0NTINUE
2185        C0N1=TBAS(NB)
2195        IF(C0N1-GWTAB(1))125,125,130
2205    125 WSP1=C0N1*AP(1,1)
2215        WSP2=C0N1*AP(1,2)
2225        G0 T0 155
2235    130 WSP1=GWTAB(1)*AP(1,1)
2245        WSP2=GWTAB(1)*AP(1,2)
2255        IF(C0N1-GWTAB(2))135,135,140
```

TABLE 32 (Cont)

```
2265   135 C0N2=C0N1-GWTAB(1)
2275       WSP1=WSP1+C0N2*AP(2,1)
2285       WSP2=WSP2+C0N2*AP(2,2)
2295       G0 T0 155
2305   140 WSP1=WSP1+(GWTAB(2)-GWTAB(1))*AP(2,1)
2315       WSP2=WSP2+(GWTAB(2)-GWTAB(1))*AP(2,2)
2325       IF(C0N1-GWTAB(3))145,145,150
2335   145 C0N2=C0N1-GWTAB(2)
2345       WSP1=WSP1+C0N2*AP(3,1)
2355       WSP2=WSP2+C0N2*AP(3,2)
2365       G0 T0 155
2375   150 C0N2=C0N1-GWTAB(3)
2385       WSP1=WSP1+(GWTAB(3)-GWTAB(2))*AP(3,1)+C0N2*AP(4,1)
2395       WSP2=WSP2+(GWTAB(3)-GWTAB(2))*AP(3,2)+C0N2*AP(4,2)
2405   155 BR(7,NB)=1.018*(WSA1+WSP1)
2415       BR(29,NB)=1.015*(WSA2+WSP2)
2425       BR(30,NB)=BR(7,NB)+BR(29,NB)
2435C---------550-10  DISPENSARY
2445       K=IBED(NB)+1
2455       C0N1=2.5*(P0(NB)*T0FF(NB)+PS(NB)*TSTU(NB))+T0FF(NB)+TSTU(NB)
2465      &+2.5*TENL(NB)*(PEE(NB)*PRE(NB)+PIE(NB)-PEE(NB)*PIE(NB))
2475       G0 T0 (160,170),K
2485   160 C0N1=3.08*C0N1
2495       BR(8,NB)=AMAX1(C0N1,3850.)
2505       G0 T0 175
2515   170 C0N1=3.696*C0N1
2525       BR(8,NB)=AMAX1(C0N1,4620.)
2535   175 C0NTINUE
2545C---------610-10  ADMIN 0FFICES
2555       BR(9,NB)=162.*TBAS(NB)*PERFAC(NB)
2565C---------711-10  FAMILY H0USING
2575       BR(10,NB)=PEE(NB)*PRE(NB)*TENL(NB)+P0(NB)* T0FF(NB)+PS(NB)*
2585&      TSTU(NB)
2595C---------000-00  INELLIGIBLE ENLISTED H0USING
2605       BR(26,NB)=(1.-PEE(NB))*TENL(NB)*PIE(NB)
2615C---------722-10  ENLISTED MENS BARRACKS
2625       BR(11,NB)=TENL(NB)*(1.-PEE(NB)*PRE(NB)-(1.-PEE(NB))*PIE(NB))
2635C---------723-10  ENLISTED MENS MESS
2645       C0N1=EMES(NB)*BR(11,NB)
2655       D0 200 I=1,7
2665       K=I
2675       IF(C0N1-FAMESS(I,1))210,210,200
2685   200 C0NTINUE
```

TABLE 32 (Cont)

```
2695  210 BR(12,NB)=C0N1*FAMESS(K,2)
2705C---------724-10  B0Q
2715        BR(13,NB)= T0FF(NB)*(1.-P0(NB))+TSTU(NB)*(1.-PS(NB))
2725C---------740-14  EXCHANGE
2735        C0N1=TBAS(NB)-TCIV(NB)
2745        IF(C0N1-EXCH(1,1))260,260,270
2755  260 BR(15,NB)=(C0N1/EXCH(1,1))*EXCH(1,2)
2765        G0 T0 320
2775  270 D0 300 I=2,10
2785        K=I
2795        IF(C0N1-EXCH(I,1))310,310,300
2805  300 C0NTINUE
2815  310 BR(15,NB)=(C0N1-EXCH(K-1,1))/(EXCH(K,1)-EXCH(K-1,1))*
2825     &(EXCH(K,2)-EXCH(K-1,2))+EXCH(K-1,2)
2835C---------740-63  SERVICE CLUB
2845  320 IF(TENL(NB)-FAEMC(1,1))330,330,340
2855  330 BR(16,NB)=(TENL(NB)/FAEMC(1,1))*FAEMC(1,2)
2865        G0 T0 370
2875  340 D0 350 I=2,8
2885        K=I
2895        IF(TENL(NB)-FAEMC(I,1))360,360,350
2905  350 C0NTINUE
2915  360 BR(16,NB)=(TENL(NB)-FAEMC(K-1,1))/(FAEMC(K,1)-FAEMC(K-1,1))
2925     &*(FAEMC(K,2)-FAEMC(K-1,2))+FAEMC(K-1,2)
2935C---------811-60
2945  370 C0NTINUE
2955C---------812-30  ELECTRIC DISTRIBUTI0N LINE
2965        BR(18,NB)=54.9*PHASP
2975C---------832-10
2985C---------842-10
2995        BR(20,NB)=-76370.+40.5*PHASP+104.8*TSTU(NB)
3005C---------851-10  R0ADS
3015        BR(21,NB)=-11.5+.0224*(TENL(NB)-PNASE(NB))
3025C---------852-10  PARKING AREA
3035        BR(22,NB)=.155*(TENL(NB)-PNASE(NB))**1.824
3045C---------124-30  READY FUEL ST0RAGE
3055        D0 400 I=1,3
3065  400 XBR2(I,NB)=(PF(NB,I)/365.)*(1.+EL(NB,I))*FUREQ(NB,I)
3075  500 C0NTINUE
3085        ISWTCH(7)=1
3095        CHAIN "PARTX*"
3105        END
```

XII. PROGRAM PARTX

PROGRAM DESCRIPTION

12.1 PROGRAM PARTX serves the purpose of providing a restart procedure within the total IFRS system flow. As a result, at any subsequent point, the user can initiate use of the IFRS model with the computations in PART6 without having to repeat all the previous computations.

12.2 The file RESTART is opened and rewound, and all variables in common are written. Then control is transferred to PART6.

12.3 A flow chart of PROGRAM PARTX is shown in Figure 12. Table 33 references the variable dictionary of PROGRAM PARTX; the program dictionary is provided in Table 34. The program listing is shown in Table 35.

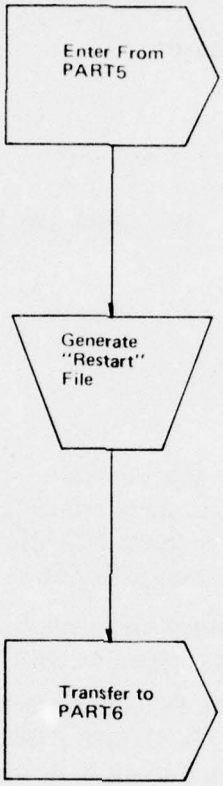


FIGURE 12. PROGRAM PARTX FLOW CHART

TABLE 33

PROGRAM PARTX VARIABLE DICTIONARY*

* See the common variable dictionary for PART2, Table 18.

TABLE 34

PROGRAM PARTX PROGRAM DICTIONARY

PARTX	Writes restart file

TABLE 35

PROGRAM PARTX LISTING

```
10000       C0MM0N IYEAR,ISWTCH(10)
10010       C0MM0N ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
10020      &ACFH(9,15),T0FF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
10030      &S0(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
10040      &IACT(25,3),ACN01(25,3),T0FF1(25),EMT1(25)
10050       C0MM0N IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
10060      &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
10070      &RL0AD(21),C0MP(21),FLCST(21),A0M(21),CNAAC(21)
10080       C0MM0N NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
10090      &PRE(9),P0(9),PS(9),PIE(9),TS(9),TH(9),TN0FF(9),TNENL(9),
10100      &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
10110       C0MM0N FAC0ST(50,6)
10120       C0MM0N FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
10130      &FAEM(8,2),TANKS(15),TAXIT0(3)
10140       C0MM0N IC0DES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
10150      &XRPI1(9,10,4),XRPI2(3,9)
10160       C0MM0N BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
10170      &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
10180      &NCAT,IYES,N0,IC0M,GT0TAL,NPH
10190       ALPHA IYES,N0,IC0M
10200       0PENFILE "RESTART"
10210       REWIND "RESTART"
10220       WRITE("RESTART")IYEAR,(ISWTCH(I),I=1,10),((ACREQ(J,K),J=1,9),
10230      &K=1,21),(TBAS(L),TNAS(L),T0FF(L),TENL(L),TSTU(L),PNASE(L),
10240      &TCIV(L),NBUSE(L),L=1,9),((BPH(M,N),M=1,9),N=1,25)
10250       WRITE("RESTART")((ASH(I,J),I=1,25),J=1,3),((ACFH(K,L),K=1,9),
10260      &L=1,15),(SI(M),S0(M),T0FF1(M),EMT1(M),M=1,25)
10270       WRITE("RESTART")((FUREQ(I,J),I=1,9),J=1,3),((PHPER(K,L),K=1,9
10280      &),L=1,5),((IACT(M,N),ACN01(M,N),M=1,25),N=1,3)
10290       WRITE("RESTART")(((RW(I,J,K),I=1,25),J=1,3),K=1,3),(IATYPE(L)
10300      &,ACA(L),ACB(L),ACC(L),ACD(L),AHM(L),ACM(L),ASM1(L),ASM2(L),
10310      &RNWYL(L),RL0AD(L),C0MP(L),FLCST(L),A0M(L),CNAAC(L),L=1,21),
10320      &((A(M,N),M=1,21),N=1,3)
10330       WRITE("RESTART")(NASNAM(I),AD(I),CU(I),IBED(I),PEE(I),
10340      &PRE(I),P0(I),PS(I),PIE(I),TS(I),TH(I),TN0FF(I),TNENL(I),
10350      &TNCIV(I),ATCF(I),PERFAC(I),EMES(I),I=1,9),((PF(J,K),EL(J,K),
10360      &J=1,9),K=1,3),((WR(L,M),L=1,9),M=1,2)
10370       WRITE("RESTART")((TENAC(I,J),I=1,9),J=1,6),((FAC0ST(K,L),K=
10380      &1,50),L=1,6),(FAPW(M),M=1,6)
10390       WRITE("RESTART")((AP(I,J),I=1,4),J=1,3),(GWTAB(K),K=1,3),
10400      &((FAMESS(L,M),L=1,7),M=1,2)
10410       WRITE("RESTART")((EXCH(I,J),I=1,10),J=1,2),((FAEM(K,L),
10420      &K=1,8),L=1,2),(TANKS(M),M=1,15),(TAXIT0(N),N=1,3)
```

TABLE 35 (Cont)

```
10430       WRITE("RESTART")(IC0DES(I),IUNITS(I),I=1,50),((IDES(J,K),
10440      &J=1,50),K=1,3),(((RPI(L,M,N),L=1,50),M=1,9),N=1,2)
10450       WRITE("RESTART")(((XRPI1(I,J,K),I=1,9),J=1,10),K=1,4),
10460      &((XRPI2(L,M),L=1,3),M=1,9)
10470       WRITE("RESTART")((BR(I,J),I=1,50),J=1,9),(((XBR1(K,L,M),
10480      &K=1,9),L=1,10),M=1,4)
10490       WRITE("RESTART")((DEF(I,J),TEX(I,J),I=1,50),J=1,9),((XBR2(K,L
10500      &),K=1,3),L=1,9),(XDEF2(M),M=1,9)
10510       WRITE("RESTART")((XDEF3(I,J),I=1,2),J=1,9),(((XDEF4(K,L,M),
10520      &K=1,3),L=1,15),M=1,9),NCAT,IYES,N0,IC0M,GT0TAL,NPH
10530       CL0SEFILE "RESTART"
10540       CHAIN "PART6*"
10550       END
```

XIII. PROGRAM PART6

PROGRAM DESCRIPTION

13.1 PROGRAM PART6[1] compares existing assets with the facilities requirements computed in PART5. If desired, the resulting excesses or deficiencies are printed, along with the available assets and requirements for each base.

13.2 Upon entering PART6, both ISWTCH(7) and ISWTCH(9) are stored so that the reading of file RESTART does not destroy any alterations in the print detail level that can be made in PART9. The file RESTART is read and, if necessary, the updated print level is recorded. The user then specifies whether he wishes to accept substandard facilities. A Yes answer still allows the choice to be made for each base; a No response is fixed for all bases.

13.3 The following procedure is completed for each base. For each facility requirement stored in the facilities requirements array BR, a comparison is made with existing assets. Any excesses noted are stored in TEX, deficiencies in DEF. Certain facilities (numbers 1, 7, 27, 29) are subtotals and the excess or deficiency is not meaningful for these categories. Thus, TEX and DEF are set to zero for these facilities.[2] Transfer is then made to Subroutine SUB4. This subroutine computes ready fuel storage, converting any deficits into number of tanks by using the tank sizes in the table TANKS.

---

[1] This program is the Excess/Deficiency Submodel discussed in Volumes I and II.

[2] Data were not available at the subtotal level for these facilities in the Assets Position Data File.

13.4 The procedure used in determining total tank requirements is to pick the largest tank size which does not exceed the deficit storage by more than half the smallest tank size. One continues to add tanks of the same or smaller size, always abiding by this criterion and always reducing the deficit until the deficit is eliminated. A deficit of less then 50 gal is ignored. The number of needed tanks of any one size is stored in array XDEF4. Once all deficits are corrected, the routine branches back to the main program.

13.5 Depending on the level of print detail, the user gets no printout or a printout of deficiencies only or has a choice of getting the total excess/deficiency table for the base considered. Included are the runway lighting and taxiway deficiencies (if any) that were previously computed in PARTY.

13.6 After printing the complete excess/deficiency table, the next base is considered. After all bases are considered, control is transferred to PART7.

13.7 A flow chart of PROGRAM PART6 is shown in Figure 13. Table 36 contains the variable dictionary of PROGRAM PART6; the program and subroutine dictionary is provided in Table 37. The program listing is shown in Table 38.

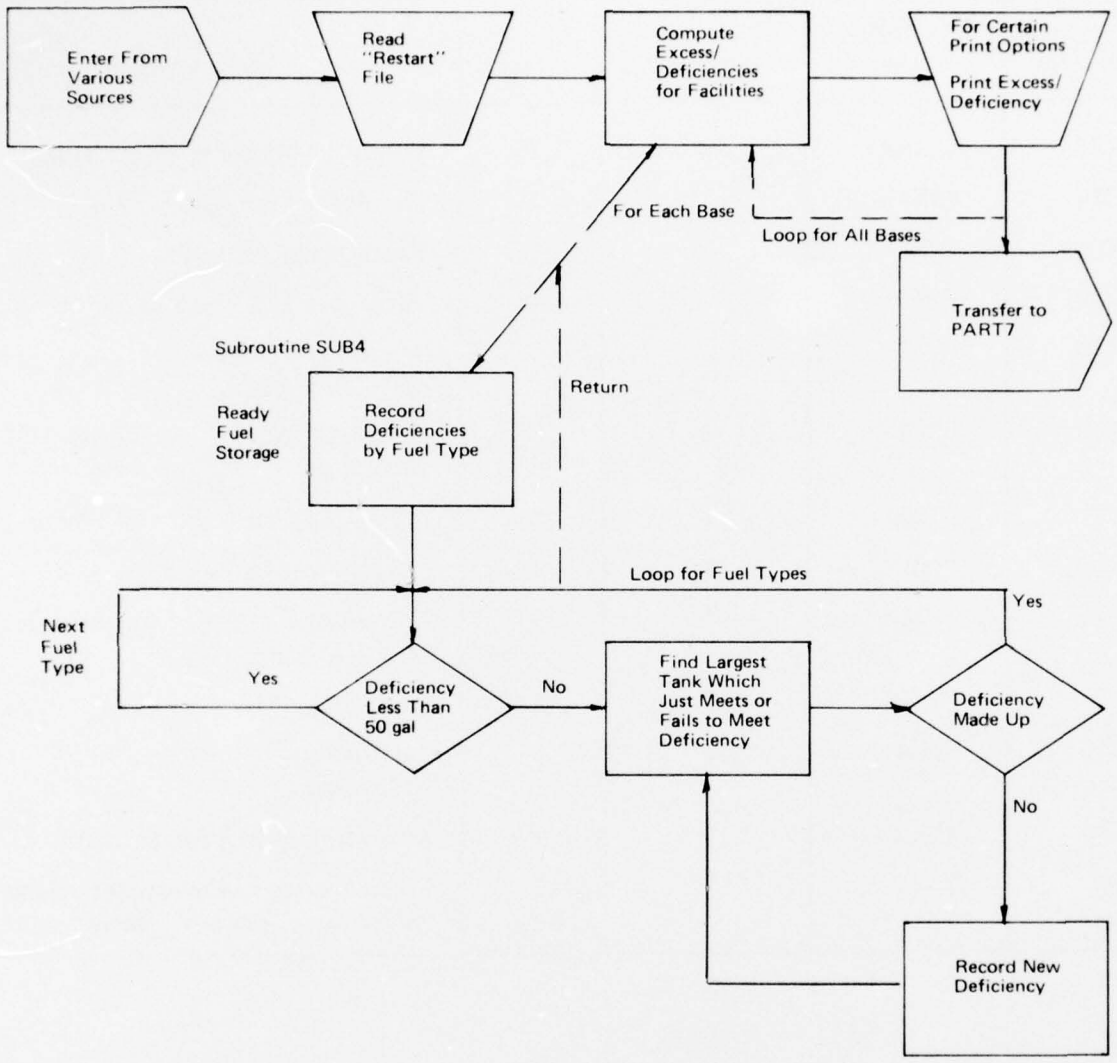


FIGURE 13. PROGRAM PART6 FLOW CHART

TABLE 36

PROGRAM PART6 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Type	Description
PART6	IARRAY	50	I	Order of print for facilities
PART6	ISAVE	1	I	Temporary storage
PART6	IKNOW	1	I	Temporary storage
PART6	IOP	1	A	Terminal Yes-No response
PART6	IOP1	1	I	Standard-substandard acceptance indicator
PART6	IOP2	1	I	Same option for all bases indicator
PART6	NSKIP	1	I	First pass through indicator
PART6	TEA	1	F	Total existing assets
PART6	TEMP	1	F	Temporary storage of excess or deficiency
PART6	LSKIP	1	I	Unused
PART6	JSKIP	1	I	Deficiency-no deficiency indicator
PART6	VAL	1	F	Adjusted value for printing
SUB4	T	3	F	Total excess or deficiency for ready fuel storage, type I

TABLE 37

PROGRAM PART6 PROGRAM AND SUBROUTINE DICTIONARY

PART6	Reads restart file and computes facility excess/deficiency
SUB4	Computes ready fuel storage excess/deficiency

TABLE 38

PROGRAM PART6 LISTING

```
1006       COMMON IYEAR,ISWTCH(10)
1016       COMMON ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
1026      &ACFH(9,15),TOFF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
1036      &SO(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
1046      &IACT(25,3),ACNO1(25,3),TOFF1(25),EMT1(25)
1056       COMMON IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
1066      &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
1076      &RLOAD(21),COMP(21),FLCST(21),AOM(21),CNAAC(21)
1086       COMMON NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
1096      &PRE(9),PO(9),PS(9),PIE(9),TS(9),TH(9),TNOFF(9),TNENL(9),
1106      &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
1116       COMMON FACOST(50,6)
1126       COMMON FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
1136      &FAEM(8,2),TANKS(15),TAXITO(3)
1146       COMMON ICODES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
1156      &XRPI1(9,10,4),XRPI2(3,9)
1166       COMMON BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
1176      &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
1186      &NCAT,IYES,NO,ICOM,GTOTAL,NPH
1196       DIMENSION IARRAY(50)
1206       ALPHA IYES,NO,ICOM,IOP
1216       DATA IARRAY/1,27,28,2,3,4,5,6,7,29,30,8,9,10,26,11,12,13,
1226      &14,15,16,17,18,19,20,21,22,23,24,25,20*0/
1236       ISAVE=ISWTCH(9)
1246       IKNOW=ISWTCH(7)
1256       OPENFILE "RESTART"
1266       REWIND "RESTART"
1276       READ ("RESTART")IYEAR,(ISWTCH(I),I=1,10),((ACREQ(J,K),J=1,9),
1286      &K=1,21),(TBAS(L),TNAS(L),TOFF(L),TENL(L),TSTU(L),PNASE(L),
1296      &TCIV(L),NBUSE(L),L=1,9),((BPH(M,N),M=1,9),N=1,25)
1306       READ ("RESTART")((ASH(I,J),I=1,25),J=1,3),((ACFH(K,L),K=1,9),
1316      &L=1,15),(SI(M),SO(M),TOFF1(M),EMT1(M),M=1,25)
1326       READ ("RESTART")((FUREQ(I,J),I=1,9),J=1,3),((PHPER(K,L),K=1,9
1336      &),L=1,5),((IACT(M,N),ACNO1(M,N),M=1,25),N=1,3)
1346       READ ("RESTART")(((RW(I,J,K),I=1,25),J=1,3),K=1,3),(IATYPE(L)
1356      &,ACA(L),ACB(L),ACC(L),ACD(L),AHM(L),ACM(L),ASM1(L),ASM2(L),
1366      &RNWYL(L),RLOAD(L),COMP(L),FLCST(L),AOM(L),CNAAC(L),L=1,21),
1376      &((A(M,N),M=1,21),N=1,3)
1386       READ ("RESTART")(NASNAM(I),AD(I),CU(I),IBED(I),PEE(I),
1396      &PRE(I),PO(I),PS(I),PIE(I),TS(I),TH(I),TNOFF(I),TNENL(I),
1406      &TNCIV(I),ATCF(I),PERFAC(I),EMES(I),I=1,9),((PF(J,K),EL(J,K),
1416      &J=1,9),K=1,3),((WR(L,M),L=1,9),M=1,2)
1426       READ("RESTART") ((TENAC(I,J),I=1,9),J=1,6),((FACOST(K,L),K=
```

TABLE 38 (Cont)

```
1436      &1,50),L=1,6),(FAPW(M),M=1,6)
1446       READ ("RESTART")(((AP(I,J),I=1,4),J=1,3),(GWTAB(K),K=1,3),
1456      &((FAMESS(L,M),L=1,7),M=1,2)
1466       READ ("RESTART")((EXCH(I,J),I=1,10),J=1,2),((FAEM(K,L),
1476      &K=1,8),L=1,2),(TANKS(M),M=1,15),(TAXIT0(N),N=1,3)
1486       READ ("RESTART")(IC0DES(I),IUNITS(I),I=1,50),((IDES(J,K),
1496      &J=1,50),K=1,3),(((RPI(L,M,N),L=1,50),M=1,9),N=1,2)
1506       READ ("RESTART")(((XRPI1(I,J,K),I=1,9),J=1,10),K=1,4),
1516      &((XRPI2(L,M),L=1,3),M=1,9)
1526       READ ("RESTART")((BR(I,J),I=1,50),J=1,9),(((XBR1(K,L,M),
1536      &K=1,9),L=1,10),M=1,4)
1546       READ ("RESTART")((DEF(I,J),TEX(I,J),I=1,50),J=1,9),((XBR2(K,L
1556      &),K=1,3),L=1,9),(XDEF2(M),M=1,9)
1566       READ ("RESTART")((XDEF3(I,J),I=1,2),J=1,9),(((XDEF4(K,L,M),
1576      &K=1,3),L=1,15),M=1,9),NCAT,IYES,N0,IC0M,GT0TAL,NPH
1586       CL0SEFILE "RESTART"
1596       IF(IKN0W.EQ.1)ISWTCH(9)=ISAVE
1606       IF((ISWTCH(9).EQ.1).0R.(ISWTCH(9).EQ.3))G0 T0 15
1616       PRINT 600
1626    15 PRINT 601
1636    16 INPUT 602,I0P
1646       IF(I0P.EQ.IYES)G0 T0 25
1656       IF(I0P.EQ.N0)G0 T0 35
1666    25 I0P1=1
1676       G0 T0 40
1686    30 PRINT 603
1696       G0 T0 16
1706    35 I0P1=0
1716       G0 T0 45
1726    40 PRINT 604
1736       INPUT 602,I0P
1746       IF(I0P.NE.IYES)G0 T0 50
1756    45 I0P2=1
1766       G0 T0 60
1776    50 IF(I0P.NE.N0)G0 T0 30
1786    55 I0P2=0
1796    60 D0 150 IB=1,9
1806       NSKIP=0
1816       IF(NBUSE(IB))80,150,80
1826    80 D0 100 L=1,NCAT
1836       IF(BR(L,IB)-.5)100,100,84
1846    84 TEX(L,IB)=0.
1856       DEF(L,IB)=0.
```

TABLE 38 (Cont)

```
1866        IF(NSKIP.EQ.1)GØ TØ 88
1876        NSKIP=1
1886        IF(IØP2)88,85,88
1896     85 PRINT 605,NASNAM(IB)
1906     91 INPUT 602,IØP
1916        IF(IØP.NE.IYES)GØ TØ 87
1926        IØP1=1
1936        GØ TØ 88
1946     87 IF(IØP.NE.NØ)GØ TØ 93
1956     92 IØP1=0
1966        GØ TØ 88
1976     93 PRINT 603
1986        GØ TØ 91
1996     88 TEA=RPI(L,IB,1)
2006        IF(IØP1)94,95,94
2016     94 TEA=TEA+RPI(L,IB,2)
2026     95 TEMP=BR(L,IB)-TEA
2036        IF(TEMP)96,100,98
2046     96 TEX(L,IB)=-TEMP
2056        GØ TØ 100
2066     98 DEF(L,IB)=TEMP
2076    100 CØNTINUE
2086        DEF(1,IB)=0.
2096        TEX(1,IB)=0.
2106        DEF(27,IB)=0.
2116        TEX(27,IB)=0.
2126        DEF(7,IB)=0.
2136        DEF(29,IB)=0.
2146        TEX(7,IB)=0.
2156        TEX(29,IB)=0.
2166        CALL SUB4(IB)
2176        IF(ISWTCH(9).EQ.1)GØ TØ 150
2186        IF(ISWTCH(9).EQ.3)GØ TØ 150
2196C--------PRINT ØUTPUT
2206        PRINT 606,NASNAM(IB)
2216        PRINT," DETAILED EXCESS-DEFICIENCY (Y,N)"
2226    111 INPUT 602,IØP
2236        IF(IØP.EQ.IYES)GØ TØ 112
2246        IF(IØP.EQ.NØ)GØ TØ 150
2256        PRINT 603
2266        GØ TØ 111
2276    112 PRINT 607
2286        DØ 140 K=1,NCAT
```

TABLE 38 (Cont)

```
2296        I=IARRAY(K)
2306        IF(BR(I,IB)-.5)140,140,130
2316   130  IF(ISWTCH(9).LT.2)GØ TØ 140
2326        IF((ISWTCH(9).EQ.4).AND.(DEF(I,IB).LT..01))GØ TØ 140
2336        PRINT 619,ICØDES(I),(IDES(I,J),J=1,3),BR(I,IB),IUNITS(I),
2346       &RPI(I,IB,1),RPI(I,IB,2),TEX(I,IB),DEF(I,IB)
2356   140  CØNTINUE
2366        PRINT,"TAXIWAYS & RUNWAY LIGHTING"
2376        IF(XDEF2(IB).LT.50.)GØ TØ 740
2386        PRINT 628,XDEF2(IB),XDEF3(1,IB),XDEF3(2,IB)
2396        GØ TØ 750
2406   740  PRINT," NØ DEFICIENCY"
2416   750  LSKIP=0
2426        JSKIP=0
2436        PRINT,"READY FUEL STØRAGE"
2446        PRINT," REQUIRED:   (THØUSANDS ØF GALS)"
2456        DØ 800 I=1,3
2466        VAL=XBR2(I,IB)
2476        IF(VAL.LT.(TANKS(1)*.5))GØ TØ 800
2486        JSKIP=1
2496        VAL=VAL/1000.
2506        GØ TØ (770,780,790),I
2516   770  PRINT 621,VAL
2526        GØ TØ 800
2536   780  PRINT 622,VAL
2546        GØ TØ 800
2556   790  PRINT 623,VAL
2566   800  CØNTINUE
2576        IF(JSKIP.EQ.0)PRINT," NØ REQUIREMENT"
2586        JSKIP=0
2596        PRINT," AVAILABLE:"
2606        DØ 810 I=1,3
2616        VAL=XRPI2(I,IB)/1000.
2626        IF(I.EQ.1)PRINT 621,VAL
2636        IF(I.EQ.2)PRINT 622,VAL
2646        IF(I.EQ.3)PRINT 623,VAL
2656   810  CØNTINUE
2666        DØ 1550 I=1,3
2676        DØ 1550 J=1,15
2686        IF(XDEF4(I,J,IB)-.01)1550,1550,1545
2696  1545  IF(JSKIP.EQ.1)GØ TØ 1546
2706        JSKIP=1
2716        PRINT," DEFICIENT"
```

TABLE 38 (Cont)

```
2726       PRINT 624
2736  1546 VAL=TANKS(J)/1000.
2746       GØ TØ (1547,1548,1549),I
2756  1547 PRINT 625,XDEF4(I,J,IB),VAL
2766       GØ TØ 1550
2776  1548 PRINT 626,XDEF4(I,J,IB),VAL
2786       GØ TØ 1550
2796  1549 PRINT 627,XDEF4(I,J,IB),VAL
2806  1550 CØNTINUE
2816       IF(JSKIP.EQ.0)PRINT," NØ DEFICIENCY"
2826   150 CØNTINUE
2836       IF(ISWTCH(9).LE.3)GØ TØ 200
2846       PRINT 608
2856   157 INPUT 602,IØP
2866       IF(IØP.EQ.IYES)GØ TØ 15
2876       IF(IØP.EQ.NØ)GØ TØ 200
2886   165 PRINT 603
2896       GØ TØ 157
2906   200 CHAIN "PART7*"
2916   600 FØRMAT(1X25HEXCESS DEFICIENCY PRØGRAM)
2926   601 FØRMAT(" ACCEPT SUBSTANDARD FACILITIES (Y,N)")
2936   602 FØRMAT(A1)
2946   603 FØRMAT(1X,"INVALID REPLY--TRY AGAIN")
2956   604 FØRMAT(1X,"SAME ØPTIØN FØR ALL BASES (Y,N)")
2966   605 FØRMAT(1X,"ACCEPT SUBSTANDARD FØR ",A4,6H (Y,N))
2976    606 FØRMAT(/1X,"NAS--",A4)
2986   607 FØRMAT(18X"REQUIRED"10X"AVAILABLE"9X"PØSITIØN"/" CØDE"
2996      &2X"DESCRIPTIØN  AMØUNT UNIT STAND. SUB-STAND. EXCESS"
3006      &2X"DEFICIENT")
3016   619 FØRMAT(1XI5,1X3A4,F8.0,1XA2,4(1XF8.0))
3026&       1XF8.0,1XF8.0,1XF8.0,1XF8.0))
3036   608 FØRMAT(1X51HDØ YØU WISH TØ MØDIFY THE SUBSTANDARD ØPTIØN (Y,N
3046      &) )
3056   621 FØRMAT(3X,"JET ",8XF7.1)
3066   622 FØRMAT(3X,"AVGAS",7XF7.1)
3076   623 FØRMAT(3X,"HELØ",8XF7.1)
3086   624 FØRMAT(2X,"NØ.",4X,"SIZE",2X,"TYPE")
3096   625 FØRMAT(1XF4.0,2XF7.0,1X,"JET")
3106   626 FØRMAT(1XF4.0,2XF7.0,1X,"AVGAS")
3116   627 FØRMAT(1XF4.0,2XF7.0,1X,"HELØ")
3126   628 FØRMAT(1X"TAXIWAY DEFICIT",2XF8.0,2X,"SY"/1X
3136      &"RUNWAY LIGHTING DEFICIT:"2XF8.0,2X"LF;"3X,F4.0,
3146      &"APPRØACH SYSTEMS")
3156   629 FØRMAT(8I8)
3166   630 FØRMAT(5E13.6)
3176   631 FØRMAT(15A4)
3186       END
```

TABLE 38 (Cont)

a. Subroutine SUBR

```
      SUBROUTINE SUB4(IB)
C--------READY FUEL STORAGE EXCESS AND DEFICIENCY
      DIMENSION T(3)
      COMMON IYEAR,ISWTCH(10)
      COMMON ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
     &ACFH(9,15),TOFF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
     &SO(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
     &IACT(25,3),ACNO1(25,3),TOFF1(25),EMT1(25)
      COMMON IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
     &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
     &RLOAD(21),COMP(21),FLCST(21),AOM(21),CNAAC(21)
      COMMON NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
     &PRE(9),PO(9),PS(9),PIE(9),TS(9),TH(9),TNOFF(9),TNENL(9),
     &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
      COMMON FACOST(50,6)
      COMMON FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
     &FAEM(8,2),TANKS(15),TAXITO(3)
      COMMON ICODES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
     &XRPI1(9,10,4),XRPI2(3,9)
      COMMON BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
     &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
     &NCAT,IYES,NO,ICOM,GTOTAL,NPH
      ALPHA IYES,NO,ICOM
      NB=IB
C--------COMPARE EXISTING CAPACITY VS REQUIREMENTS
      DO 10 I=1,3
   10 T(I)=XRPI2(I,NB)-XBR2(I,NB)
      DO 16 I=1,3
      DO 15 J=1,15
   15 XDEF4(I,J,NB)=1.E-6
   16 CONTINUE
C---------FIND NUMBER OF TANKS IN TABLE
      DO 20 J=1,15
     JSUP=J-1
      IF(TANKS(J)-.01)25,25,20
   20 CONTINUE
      JSUP=15
C---------COMPUTE NO. OF TANKS REQUIRED BY FUEL TYPE
   25 DO 50 I=1,3
C---------IF DEFICIENCY LT 50, FORGET IT
      IF(T(I)+50.)30,50,50
   30 DO 35 J=1,JSUP
```

TABLE 38 (Cont)

a. Subroutine SUB4 (Cont)

```
3616        K=J-1
3626        C=T(I)+TANKS(J)
3636        IF(C)35,38,38
3646     35 CØNTINUE
3656        K=JSUP
3666     36 XDEF4(I,K,NB)=XDEF4(I,K,NB)+1.
3676        T(I)=T(I)+TANKS(K)
3686        GØ TØ 30
3696     38 IF(K)39,39,43
3706     39 K=1
3716     40 XDEF4(I,K,NB)=XDEF4(I,K,NB)+1.
3726        GØ TØ 50
3736     43 IF(C-(TANKS(1)/2.))44,44,36
3746     44 K=K+1
3756        GØ TØ 40
3766     50 CØNTINUE
3776        RETURN
3786        END
```

XIV. PROGRAM PART7

PROGRAM DESCRIPTION

14.1 PROGRAM PART7[1/] computes facility and aircraft investment costs. Upon entering PART7, the constant C is set to 1,000 so that later costs can be converted to thousands of dollars. The variable TCOSTI, which is used to accumulate total investment cost, is set to zero.

14.2 For print detail levels greater than 2 (entered in PART5), the base name is printed as a heading for the investment cost breakdown or base total. The investment cost for each facility is then computed where, if appropriate, control is transferred to Subroutine SIZE to compute a cost variance factor, which depends on the difference between the required and standard size for the particular facility.

14.3 For print level 5, the user is asked if he wishes a detailed breakdown of facility investment costs. If the response is Yes, the breakdown is printed, including the facilities: ready fuel storage, runway lighting, and taxiways. Finally, the base total investment cost is displayed (excluding runway costs at this time). Also for print level 5, the user has the option to avoid meeting all deficits. If he chooses to delete certain investments, the results of his decisions are recorded, the costs subtracted, and the new base total investment cost is displayed. As the costs are summed, the asset position arrays within the IFRS system are updated to temporarily incorporate the new facilities in the array. Once all bases have been considered, the total facility investment cost for all bases is printed. Control is then transferred to Subroutine ICOSTAC for the computation of aircraft investment costs.

1/ This program is part of the Total Systems Cost Submodel discussed in Volumes I and II.

14.4 In ICOSTAC, the required aircraft are summed by type over all bases. This total is then compared with the existing aircraft inventory. Where there are deficiencies, investment cost is computed. A summary (total deficient aircraft and total cost) or detailed print out (assets position and costs by type aircraft) is printed as called for by the level of print detail. If aircraft investment cost exceeds $5,000, the option is given to return to the LSR Generator to constrain output. If the response is Yes, control is transferred to PART8; with a No response, the aircraft asset position array (CNAAC) is updated and the routine transfers back to the main program where control is then transferred to PART9.

14.5 A flow chart of PROGRAM PART7 is shown in Figure 14. Table 39 contains the variable dictionary of PROGRAM PART7; the program and subroutine dictionary is provided in Table 40. The program listing is shown in Table 41.

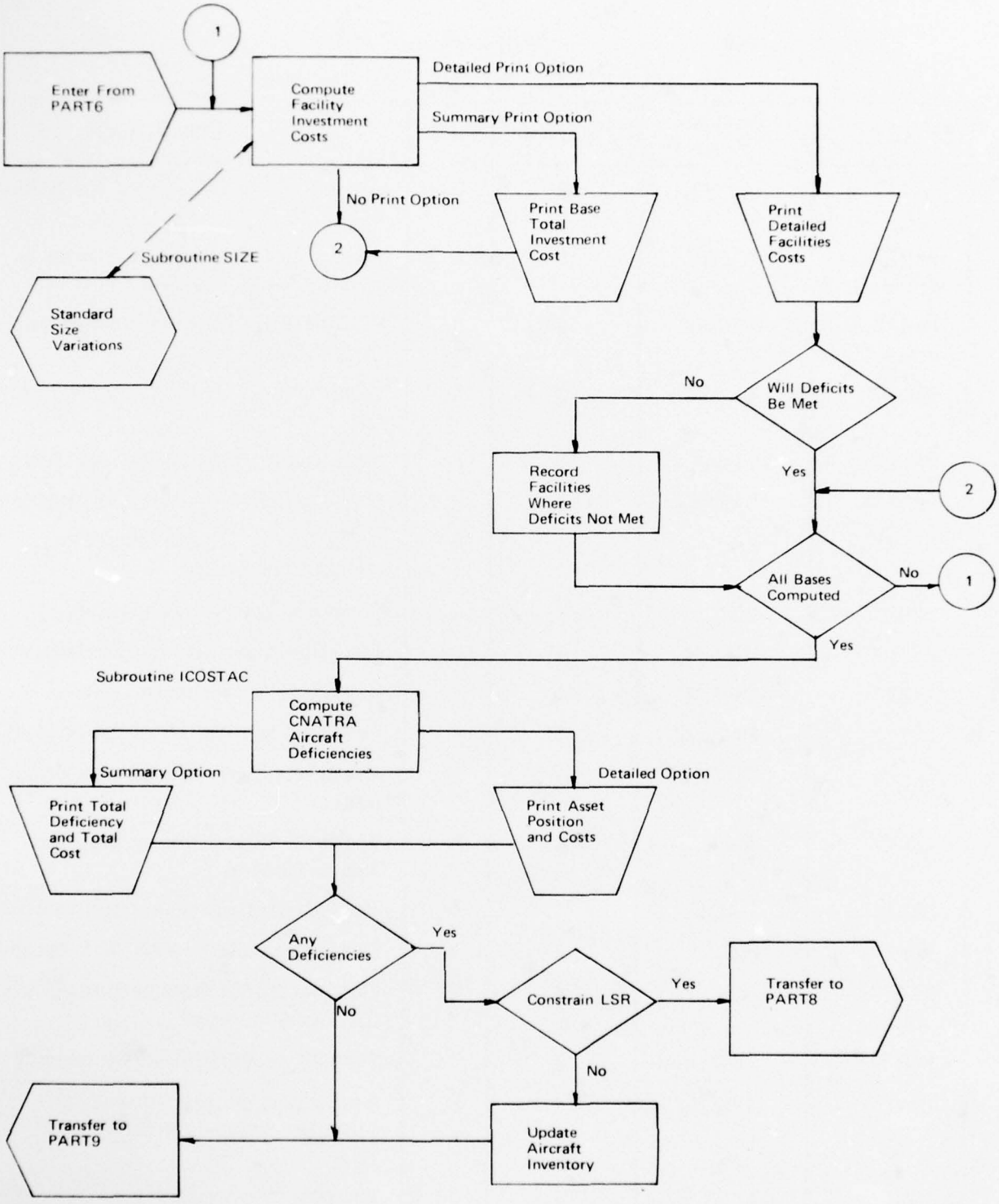


FIGURE 14. PROGRAM PART7 FLOW CHART

TABLE 39

PROGRAM PART7 VARIABLE DICTIONARY

Location	Variable Name	Dimension	Type	Description
PART7	IARRAY	50	I	See variable dictionary for PART6, Table 36
PART7	TCOST	15	F	Ready fuel tank cost for size I, thousands of dollars
PART7	TCOSTI	1	F	Total facilities investment cost for all bases
PART7	PCOSTI	1	F	Base total facilities investment cost
PART7	COST	50	F	Investment cost for I^{th} facility
PART7	NSKIP	50	I	Deficiency I met-not met indicator
PART7	Y	1	F	Relative-to-standard size cost adjustment factor
PART7	IOP	1	A	Terminal Yes-No response
PART7	JUMP	1	I	Detailed cost breakdown indicator
PART7	COST1	1	F	Taxiways investment cost
PART7	COST2	1	F	Runway lighting investment cost
PART7	COST3	1	F	Ready fuel storage investment cost
PART7	ITNK	1	I	Ready fuel deficiency met-not met indicator
PART7	IMOD	1	I	Not-all-deficiencies met indicator
PART7	NN	1	I	Category code, input from terminal
PART7	IC	1	A	Common input from terminal, used for format checks
PART7	XX	1	F	Percent of deficiency to be met
PART7	TKSUM	1	F	Ready fuel storage capacity increase during current model year
SIZE	X	1	F	Temporary storage
ICOSTAC	TOTAC	20	F	Total aircraft, type I, required for all bases

TABLE 39 (Cont)

Location	Variable Name	Dimension	Type	Description
ICOSTAC	JUMP	1	I	Detailed or summary print indicator
ICOSTAC	IOP	1	A	Terminal Yes-No response
ICOSTAC	DEFAC1	1	F	Deficient aircraft
ICOSTAC	DEFAC	1	F	Total deficient aircraft, all types
ICOSTAC	COST1	1	F	Investment cost for an aircraft type
ICOSTAC	COST2	1	F	Aircraft support cost for an aircraft type
ICOSTAC	COST	1	F	Total aircraft investment for an aircraft type
ICOSTAC	TCOST	1	F	Total aircraft investment for all aircraft types

TABLE 40

PROGRAM PART7 PROGRAM AND SUBROUTINE DICTIONARY

PART7	Computes and prints investment costs for facilities
SIZE	Computes standard size adjustment factor
ICOSTAC	Computes aircraft deficiencies and investment costs

TABLE 41

PROGRAM PART7 LISTING

```
      C0MM0N IYEAR,ISWTCH(10)
      C0MM0N ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
     &ACFH(9,15),T0FF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
     &S0(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
     &IACT(25,3),ACN01(25,3),T0FF1(25),EMT1(25)
      C0MM0N IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
     &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
     &RL0AD(21),C0MP(21),FLCST(21),A0M(21),CNAAC(21)
      C0MM0N NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
     &PRE(9),P0(9),PS(9),PIE(9),TS(9),TH(9),TN0FF(9),TNENL(9),
     &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
      C0MM0N FAC0ST(50,6)
      C0MM0N FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
     &FAEM(8,2),TANKS(15),TAXIT0(3)
      C0MM0N IC0DES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
     &XRPI1(9,10,4),XRPI2(3,9)
      C0MM0N BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
     &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
     &NCAT,IYES,N0,IC0M,GT0TAL,NPH
      DIMENSI0N IARRAY(50),TC0ST(15),C0ST(50),NSKIP(50)
      ALPHA IYES,N0,IC0M,I0P,IC
      DATA IARRAY/1,27,28,2,3,4,5,6,7,29,30,8,9,10,26,11,12,13,
     &14,15,16,17,18,19,20,21,22,23,24,25,20*0/
      DATA TC0ST/1.2,2.3,4.2,5.7,8.5,9.6,14.,18.,25.,40.5,
     &57.,58.,76.5,92.,102.5/
      C=1000.
      TC0STI=0.
      IF(ISWTCH(9).LT.3)G0 T0 3
      PRINT 602
    3 D0 100 IB=1,9
      PC0STI=0.
      IF(NBUSE(IB))4,100,4
    4 D0 5 I=1,50
      C0ST(I)=0.
    5 NSKIP(I)=0
      IF(ISWTCH(9).LT.3)G0 T0 6
      PRINT 601,NASNAM(IB)
    6 C0ST(1)=(DEF(1,IB)*FAC0ST(1,1)*.95)/C
      C0ST(2)=DEF(2,IB)*FAC0ST(2,1)*1.0575*FAC0ST(2,4)*(.95/C)
      CALL SIZE(3,IB,Y)
      C0ST(3)=DEF(3,IB)*FAC0ST(3,1)*Y*(FAC0ST(3,2)+.0575)*.95*
     &(FAC0ST(3,4)/C)
      CALL SIZE(4,IB,Y)
      C0ST(4)=DEF(4,IB)*FAC0ST(4,1)*Y*(FAC0ST(4,2)+.0575)*.95*
     &(FAC0ST(4,4)/C)
```

TABLE 41 (Cont)

```
1907        CALL SIZE(5,IB,Y)
1927        C0ST(5)=DEF(5,IB)*FAC0ST(5,1)*Y*(FAC0ST(5,2)+.0575)*.95*
1947&       (FAC0ST(5,4)/C)
1967        CALL SIZE(6,IB,Y)
1987        C0ST(6)=DEF(6,IB)*FAC0ST(6,1)*Y*(FAC0ST(6,2)+.0575)*.95*
2007&       (FAC0ST(6,4)/C)
2027        C0ST(7)=DEF(7,IB)*FAC0ST(7,1)*(FAC0ST(7,2)+.0575)*(.95/C)
2047        CALL SIZE(8,IB,Y)
2067        C0ST(8)=DEF(8,IB)*FAC0ST(8,1)*Y*(FAC0ST(8,2)+.0575)*.95*
2087&       (FAC0ST(8,4)/C)
2107        CALL SIZE(9,IB,Y)
2127        C0ST(9)=DEF(9,IB)*FAC0ST(9,1)*Y*(FAC0ST(9,2)+.0575)*.95*
2147&       (FAC0ST(9,4)/C)
2167        C0ST(10)=DEF(10,IB)*FAC0ST(10,1)*(.95/C)
2187        C0ST(11)=DEF(11,IB)*FAC0ST(11,1)*FAC0ST(11,2)*(.95/C)
2207        CALL SIZE(12,IB,Y)
2227        C0ST(12)=DEF(12,IB)*FAC0ST(12,1)*Y*(FAC0ST(12,2)+.0575)*.95*
2247&       (FAC0ST(12,4)/C)
2267        C0ST(13)=DEF(13,IB)*FAC0ST(13,1)*FAC0ST(13,2)*(.95/C)
2287        CALL SIZE(15,IB,Y)
2307        C0ST(15)=DEF(15,IB)*FAC0ST(15,1)*Y*(FAC0ST(15,2)+.0575)*.95*
2327&       (FAC0ST(15,4)/C)
2347        CALL SIZE(16,IB,Y)
2367        C0ST(16)=DEF(16,IB)*FAC0ST(16,1)*Y*(FAC0ST(16,2)+.0575)*.95*
2387&       (FAC0ST(16,4)/C)
2407        C0ST(17)=DEF(17,IB)*FAC0ST(17,1)*1.0575*.95*(FAC0ST(17,4)/C)
2427        C0ST(18)=DEF(18,IB)*FAC0ST(18,1)*1.0575*.95*(FAC0ST(18,4)/C)
2447        C0ST(21)=DEF(21,IB)*FAC0ST(21,1)*1.0575*.95*(FAC0ST(21,4)/C)
2467        C0ST(22)=DEF(22,IB)*FAC0ST(22,1)*1.0575*.95*(FAC0ST(22,4)/C)
2487        C0ST(25)=DEF(25,IB)*FAC0ST(25,1)*1.0575*.95*(FAC0ST(25,4)/C)
2507        C0ST(27)=DEF(27,IB)*FAC0ST(27,1)*(.95/C)
2527        C0ST(28)=DEF(28,IB)*FAC0ST(28,1)*(.95/C)
2547        C0ST(29)=DEF(29,IB)*FAC0ST(29,1)*(FAC0ST(29,2)+.06)*(.95/C)
2567        C0ST(30)=DEF(30,IB)*FAC0ST(30,1)*(FAC0ST(30,2)+.0575)
2587       &*(.95/C)
2607        IF(ISWTCH(9).NE.5)G0 T0 45
2627        PRINT," DETAILED BREAKD0WN (Y,N)"
2647      8 INPUT,I0P
2667        JUMP=3
2687        IF(I0P.EQ.IYES)JUMP=0
2707        IF(I0P.EQ.N0)JUMP=1
2727        IF(JUMP.NE.3)G0 T0 10
2747        PRINT 607
2767        G0 T0 8
```

TABLE 41 (Cont)

```
2787    10 D0 30 K=1,NCAT
2807       I=IARRAY(K)
2827       IF(JUMP.EQ.1)G0 T0 25
2847       IF(C0ST(I)-.5)30,30,20
2867    20 PRINT 600,IC0DES(I),(IDES(I,J),J=1,3),C0ST(I)
2887    25 PC0STI=PC0STI+C0ST(I)
2907    30 C0NTINUE
2927       C0ST1=XDEF2(IB)*6.40*(.95/C)
2947       IF(JUMP.EQ.1)G0 T0 32
2967       IF(C0ST1.LT..5)G0 T0 32
2987       PRINT 612,C0ST1
3007    32 PC0STI=PC0STI+C0ST1
3027       C0ST2=(XDEF3(1,IB)*34.50+XDEF3(2,IB)*69000.)*1.0575*(.95/C)
3047       IF(JUMP.EQ.1)G0 T0 34
3067       IF(C0ST2.LT..5)G0 T0 34
3087       PRINT 613,C0ST2
3107    34 PC0STI=PC0STI+C0ST2
3127       X=0.
3147       C0ST3=0.
3167       D0 36 I=1,15
3187       D0 35 J=1,3
3207    35 X=X+XDEF4(J,I,IB)
3227    36 C0ST3=C0ST3+X*TC0ST(I)
3247       ITNK=0
3267       IF(JUMP.EQ.1)G0 T0 37
3287       IF(C0ST3.LT..5)G0 T0 37
3307       PRINT 614,C0ST3
3327    37 PC0STI=PC0STI+C0ST3
3347       IF(ISWTCH(9).LT.3)G0 T0 90
3367       PRINT 603,PC0STI
3387       IM0D=0
3407       IF(PC0STI-.5 )100,100,50
3427    45 JUMP=1
3447       G0 T0 10
3467    50 IF(ISWTCH(9).NE.5)G0 T0 90
3487       IF(JUMP.EQ.1)G0 T0 90
3507       PRINT 605
3527    55 INPUT 606,I0P
3547       IF(I0P.EQ.IYES)G0 T0 90
3567       IF(I0P.EQ.N0)G0 T0 70
3587    65 PRINT 607
3607       G0 T0 55
```

TABLE 41 (Cont)

```
3627    70 PRINT 608
3647       IM0D=1
3667    72 INPUT 609,NN,IC,XX
3687       IF(NN)75,90,75
3707    75 IF(IC0M.E0.IC)G0 T0 77
3727    76 PRINT 607
3747       G0 T0 72
3767    77 D0 80 J=1,NCAT
3787      IF(NN-IC0DES(J))80,78,80
3807    78 K=J
3827       KN=0
3847       G0 T0 85
3867    80 C0NTINUE
3887       IF(NN.EQ.11210)G0 T0 120
3907       IF(NN.EQ.13630)G0 T0 130
3927       IF(NN.EQ.12430)G0 T0 140
3947       KN=0
3967       G0 T0 76
3987    85 IF(XX-1.0)87,87,86
4007    86 PRINT 610
4027       G0 T0 72
4047    87 IF(KN.GT.0)G0 T0 88
4067       NSKIP(K)=1
4087       RPI (K,IB,1)=RPI (K,IB,1)+XX*DEF(K,IB)
4107       PC0STI=PC0STI-(1.-XX)*C0ST(K)
4127   875 PRINT 611
4147       G0 T0 72
4167    88 G0 T0 (881,882,883),KN
4187   881 PC0STI=PC0STI-(1.-XX)*C0ST1
4207       G0 T0 875
4227   882 PC0STI=PC0STI-(1.-XX)*C0ST2
4247       G0 T0 875
4267   883 PC0STI=PC0STI-(1.-XX)*C0ST3
4287       ITNK=1
4307       D0 885 I=1,3
4327       TKSUM=0.
4347       D0 884 J=1,15
4367   884 TKSUM=TKSUM+TANKS(J)*XDEF4(I,J,IB)
4387       TKSUM=TKSUM*XX
4407   885 XRPI2(I,IB)=XRPI2(I,IB)+TKSUM
4427       G0 T0 875
4447    90 D0 95 I=1,NCAT
4467       IF(NSKIP(I)-1)92,95,92
4487    92 RPI (I,IB,1)=RPI (I,IB,1)+DEF(I,IB)
4507    95 C0NTINUE
```

TABLE 41 (Cont)

```
4527        IF(ITNK.EQ.1)G0 T0 98
4547        D0 97 I=1,3
4567        TKSUM=0.
4587        D0 96 J=1,15
4607     96 TKSUM=TKSUM+TANKS(J)*XDEF4(I,J,IB)
4627     97 XRPI2(I,IB)=XRPI2(I,IB)+TKSUM
4647     98 IF(IM0D.EQ.0)G0 T0 100
4667        PRINT 615,PC0STI
4687    100 TC0STI=TC0STI+PC0STI
4707        IF(ISWTCH(9).LT.3)G0 T0 110
4727        PRINT 604,IYEAR,TC0STI
4747    110 GT0TAL=GT0TAL+TC0STI
4767        CALL IC0STAC
4787        CHAIN "PART9*"
4807    120 KN=1
4827        G0 T0 85
4847    130 KN=2
4867        G0 T0 85
4887    140 KN=3
4907        G0 T0 85
4927    600 F0RMAT(1XI5,1X3A4,1XF8.1)
4947    601 F0RMAT(1X5HNAS-- A4,/3X10HFACILITIES )
4967    602 F0RMAT(//1X26HINVESTMENT C0ST (TH0USANDS /22X11H0F D0LLARS) )
4987    603 F0RMAT(/9X10HBASE T0TALF9.1)
5007    604 F0RMAT(/8X"NAS T0TAL"/1X"YEAR "I4,9XF9.1)
5027    605 F0RMAT(1X30HWILL ALL DEFICITS BE MET (Y,N) )
5047    606 F0RMAT(A1)
5067    607 F0RMAT(1X24HINVALID REPLY--TRY AGAIN )
5087    608 F0RMAT(1X61HWRITE CATEG0RY C0DE AND PERCENT 0F DEFICIT T0 BE
5107       &MADE
5127       &UP AS--/1X9HNNNNNN,.XX/1X54H(NEED 0NLY ENTER CASES WHERE PERCE
5147       &NT IS
5167       & LESS THAN 1.0)/1X31HUSE CATEG0RY C0DES GIVEN AB0VE,/1X41HZER
5187       &0 CAT
5207       &EG0RY C0DE INDICATES END 0F INPUT )
5227    609 F0RMAT(I5,A1,F3.0)
5247    610 F0RMAT(1X40HPERCENT MUST BE LESS THAN 1.0--TRY AGAIN )
5267    611 F0RMAT(1X6HNEXT--)
5287    612 F0RMAT(" 11210",1X"TAXIWAYS     ",1XF8.1)
5307    613 F0RMAT(" 13630",1X"RUNWY LIGHTS",1XF8.1)
5327    614 F0RMAT(" 12430",1X"RDY FUEL STG",1XF8.1)
5347    615 F0RMAT(/5X"NEW BASE T0TAL"F9.1)
5367        END
```

TABLE 41 (Cont)

a. Subroutine SIZE

```
5387        SUBROUTINE SIZE(I,IB,Y)
5407        COMMON IYEAR,ISWTCH(10)
5427        COMMON ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
5447       &ACFH(9,15),TOFF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
5467       &SO(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
5487       &IACT(25,3),ACNO1(25,3),TOFF1(25),EMT1(25)
5507        COMMON IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
5527       &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
5547       &RLOAD(21),COMP(21),FLCST(21),AOM(21),CNAAC(21)
5567        COMMON NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
5587       &PRE(9),PO(9),PS(9),PIE(9),TS(9),TH(9),TNOFF(9),TNENL(9),
5607       &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
5627        COMMON FACOST(50,6)
5647        COMMON FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
5667       &FAEM(8,2),TANKS(15),TAXITO(3)
5687        COMMON ICODES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
5707       &XRPI1(9,10,4),XRPI2(3,9)
5727        COMMON BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
5747       &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
5767       &NCAT,IYES,NO,ICOM,GTOTAL,NPH
5787        ALPHA IYES,NO,ICOM,IOP
5807        X=DEF(I,IB)
5827        IF(X-.5)10,10,20
5847    10  Y=0.
5867        RETURN
5887    20  IF(FACOST(I,3).LT..01)GO TO 40
5907        X=X/FACOST(I,3)
5927        IF(X-1.)30,40,50
5947    30  Y=1.007/(X**.10085)
5967        RETURN
5987    40  Y=1.
6007        RETURN
6027    50  Y=1.0012/(X**.06329)
6047        RETURN
6067        END
```

TABLE 41 (Cont)

b. Subroutine ICOSTAC

```
6087        SUBR0UTINE IC0STAC
6107        C0MM0N IYEAR,ISWTCH(10)
6127        C0MM0N ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
6147       &ACFH(9,15),T0FF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
6167       &S0(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
6187       &IACT(25,3),ACN01(25,3),T0FF1(25),EMT1(25)
6207        C0MM0N IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
6227       &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
6247       &RL0AD(21),C0MP(21),FLCST(21),A0M(21),CNAAC(21)
6267        C0MM0N NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
6287       &PRE(9),P0(9),PS(9),PIE(9),TS(9),TH(9),TN0FF(9),TNENL(9),
6307       &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
6327        C0MM0N FAC0ST(50,6)
6347        C0MM0N FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
6367       &FAEM(8,2),TANKS(15),TAXIT0(3)
6387        C0MM0N IC0DES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
6407       &XRPI1(9,10,4),XRPI2(3,9)
6427        C0MM0N BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
6447       &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
6467       &NCAT,IYES,N0,IC0M,GT0TAL,NPH
6487        DIMENSI0N T0TAC(20)
6507        ALPHA I0P,IYES,N0
6527        D0 10 J=1,15
6547        T0TAC(J)=0.
6567        D0 10 I=1,9
6587     10 T0TAC(J)=T0TAC(J)+ACREQ(I,J)
6607        D0 12 J=1,15
6627     12 T0TAC(J)=T0TAC(J)*1.15
6647        TC0ST=0.
6667        DEFAC=0.
6687        JUMP=0
6707        IF(ISWTCH(9).EQ.3)JUMP=1
6727        IF(ISWTCH(9).EQ.4)JUMP=1
6747        IF(ISWTCH(9).NE.5)G0 T0 25
6767        PRINT,"


                  DETAILED A/C ASSET P0SITI0N & INVESTMENT (Y,N)"
6787     15 INPUT,I0P
6807        IF(I0P.EQ.IYES)G0 T0 20
6827        IF(I0P.EQ.N0)G0 T0 18
6847        PRINT,"INVALID REPLY--RETYPE"
6867        G0 T0 15
```

TABLE 41 (Cont)

b. Subroutine ICOSTAC (Cont)

```
6887    18 JUMP=1
6907    20 IF(JUMP.EQ.0)PRINT 600
6927    25 DO 30 I=1,15
6947       IF(TOTAC(I).LT..5)GO TO 30
6967       DEFAC1=TOTAC(I)-CNAAC(I)
6987       IF(DEFAC1.LT.0.)DEFAC1=0.
7007       DEFAC=DEFAC+DEFAC1
7027       COST1=DEFAC1*FLCST(I)
7047       COST2=COST1*.15
7067       COST=COST1+COST2
7087       TCOST=TCOST+COST
7107       IF(ISWTCH(9).LT.3)GO TO 30
7127       IF(JUMP.EQ.1)GO TO 30
7147       PRINT 601,IATYPE(I),CNAAC(I),TOTAC(I),DEFAC1,COST1,COST2,
7167      &COST
7187    30 CONTINUE
7207       GTOTAL=GTOTAL+TCOST
7227       IF(ISWTCH(9).LT.3)GO TO 80
7247       IF(JUMP.EQ.0)GO TO 70
7267       PRINT,"A/C INVESTMENT (CNATRA)---SUMMARY"
7287       PRINT 602,DEFAC,TCOST
7307       GO TO 70
7327    50 IF(TCOST.LT.5.)GO TO 80
7347    60 PRINT,"

                DO YOU WISH TO CONSTRAIN LSR OUTPUT (Y,N)"
7367    61 INPUT,IOP
7387       IF(IOP.EQ.IYES)CHAIN "PART8*"
7407       IF(IOP.EQ.NO)GO TO 80
7427       PRINT,"BAD REPLY--RETYPE"
7447       GO TO 61
7467    70 IF(ISWTCH(9).NE.5)GO TO 80
7487       GO TO 50
7507    80 DO 81 I=1,15
7527    81 CNAAC(I)=AMAX1(CNAAC(I),TOTAC(I))
7547       RETURN
7567   600 FORMAT(1X"A/C INVESTMENT & ASSET POSITION---CNATRA"/
7587      &7X"ASSET  POSITION"9X6(1H-)" COSTS (THOUS.) "4(1H-)/
7607      &2X"A/C AVAILABLE REQ'D  DEFICIT FLYAWAY  SUPPORT   TOTAL")
7627   601 FORMAT(1XA4,F8.0,F9.0,F8.0,3F9.0)
7647   602 FORMAT(2X"TOTAL A/C  TOTAL"/2X"DEFICIENT   COST (THOUS.)"/
7667      &3XF7.0,6XF8.0)
7687       END
```

XV. PROGRAM PART8

PROGRAM DESCRIPTION

15.1 PROGRAM PART8 is entered only if the user has chosen to return from PART4 or PART7 to constrain LSR output. It writes out on file RETURN1, the current phase to base allocation, so that it may be recovered upon re-entering PART2.

15.2 Upon entering PART8, ISWTCH(10) is set to 2, which is a signal to the LSR Generator that the user wishes to constrain LSR outputs. Then the file RETURN1 is rewound and the present phase to base allocation information is stored. Finally, control is transferred to LSRM.

15.3 A flow chart of PROGRAM PART8 is shown in Figure 15. Table 42 references the variable dictionary of PROGRAM PART8; the program and subroutine dictionary is provided in Table 43. The program listing is shown in Table 44.

FIGURE 15. PROGRAM PART8 FLOW CHART

TABLE 42

PROGRAM PART8 VARIABLE DICTIONARY*

*See the common variable dictionary for PART2, Table 18.

TABLE 43

PROGRAM PART8 PROGRAM DICTIONARY

PART8	Writes Return files in preparation for transferring to the LSR Generator for constraining output

TABLE 44

PROGRAM PART8 LISTING

```
1008       C0MM0N IYEAR,ISWTCH(10)
1028       C0MM0N ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
1048      &ACFH(9,15),T0FF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
1068      &S0(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
1088      &IACT(25,3),ACN01(25,3),T0FF1(25),EMT1(25)
1108       C0MM0N IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
1128      &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
1148      &RL0AD(21),C0MP(21),FLCST(21),A0M(21),CNAAC(21)
1168       C0MM0N NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
1188      &PRE(9),P0(9),PS(9),PIE(9),TS(9),TH(9),TN0FF(9),TNENL(9),
1208      &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
1228       C0MM0N FAC0ST(50,6)
1248       C0MM0N FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
1268      &FAEM(8,2),TANKS(15),TAXIT0(3)
1288       C0MM0N IC0DES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
1308      &XRPI1(9,10,4),XRPI2(3,9)
1328       C0MM0N BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
1348      &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
1368      &NCAT,IYES,N0,IC0M,GT0TAL,NPH
1388       ISWTCH(10)=2
1408       0PENFILE "RETURN1"
1428       REWIND "RETURN1"
1448       WRITE("RETURN1",604)BPH
1468       WRITE("RETURN1",602)NBUSE
1488       CL0SEFILE "RETURN1"
1508       CHAIN "XLSRM*"
1528   600 F0RMAT(A4)
1548   601 F0RMAT(I5,E13.6)
1568   602 F0RMAT(8I8)
1588   603 F0RMAT(15A4)
1608   604 F0RMAT(5E13.6)
1628       END
```

XVI. PROGRAM PART9

PROGRAM DESCRIPTION

16.1 PROGRAM PART9 [1/] computes operations and maintenance (O&M) costs and combines these costs with investment costs, which were computed in PART5 and PART7, to arrive at the total systems cost.

16.2 Upon entering PART9, ISA is set to 0 if no O&M printout is to be displayed (print level 1 or 2), 1 if only a summary printout (print level 3 or 4), or 2 if the complete breakdown of O&M costs is desired. If the print level is 5, the user makes the choice between detailed or summary printout.

16.3 Storage locations SUB1, SUB2, and SUB3, in which cost totals will be stored, are first initialized to zero. Then the following O&M costs are computed by base: first, facility O&M costs; then runway O&M costs, and finally, personnel costs. Next, fuel costs are computed, followed by aircraft O&M costs, which utilize the annual flight hours per aircraft type as they were accumulated in PART3. Finally, base support costs are computed.

16.4 If details had been requested previously, these costs would have been printed out as they were computed. Otherwise, a summary O&M cost table is printed at this point. The computations proceed for all bases after which the total systems costs, including total investment cost, O&M costs, and those fixed costs ($6.2 milion) in the model are computed and displayed.

1/ This program is part of the Total Systems Cost Submodel discussed in Volumes I and II.

16.5 If the user wishes to return to the excess deficiency program as it was when it was last entered (with possibly a new level of print detail), he so indicates and control is transferred to PART6. If not, then the user may choose to go on and generate another LSR for the next simulated year; in which case, the date and year count are updated and the file RETURN, which contains information on the asset position that will be required upon returning to PART2, is written. If neither option is taken, the program ends at line 3549.

16.6 A flow chart of PROGRAM PART9 is shown in Figure 16. Table 45 contains the variable dictionary of PROGRAM PART9; the program and subroutine dictionary is provided in Table 46. The program listing is shown in Table 47.

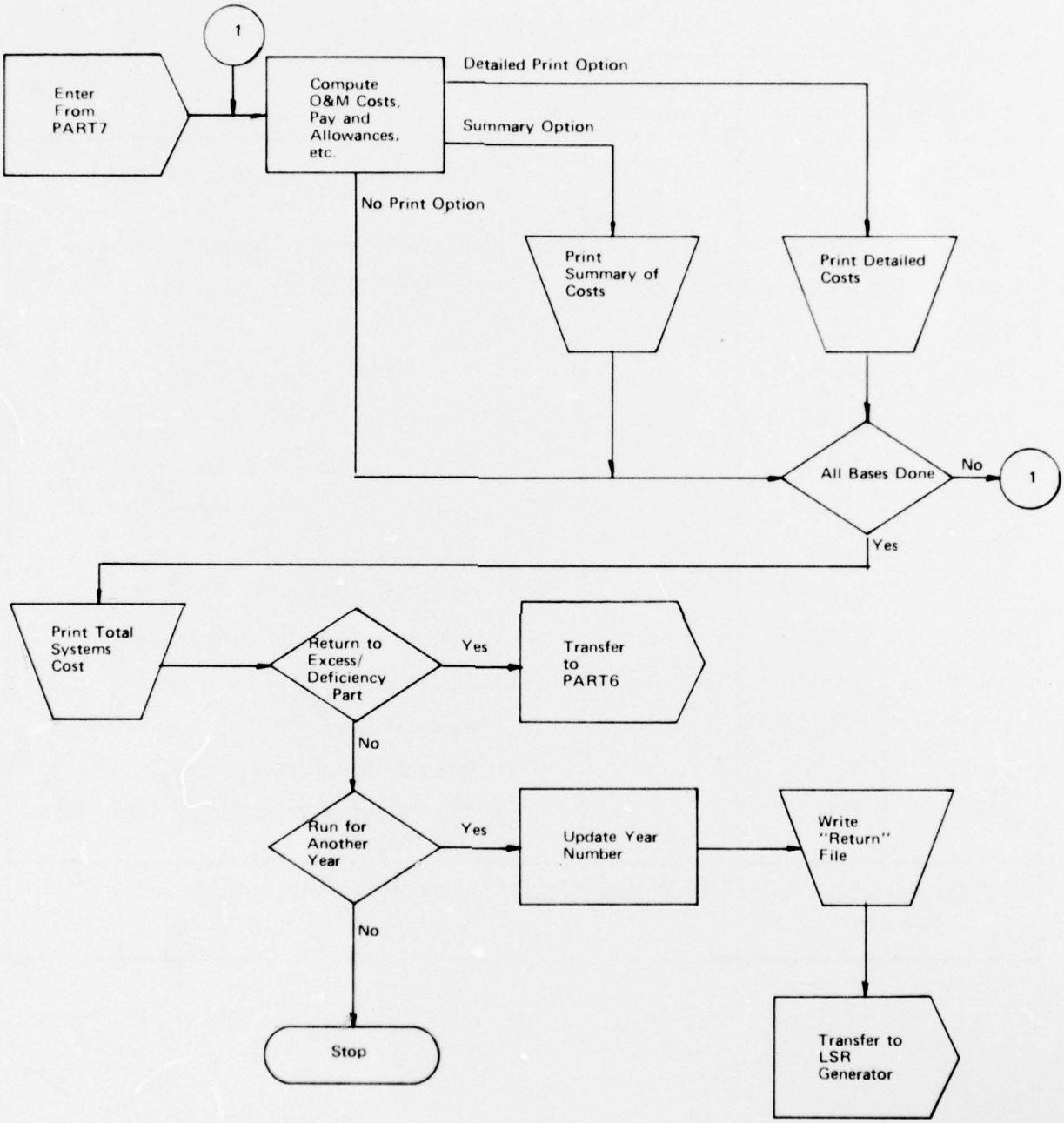


FIGURE 16. PROGRAM PART9 FLOW CHART

TABLE 45

PROGRAM PART9 VARIABLE DICTIONARY*

Location	Variable Name	Dimension	Type	Description
PART9	IARRAY	50	I	See common variable description for PART6, Table 36
PART9	ISA	1	I	Detail or summary print indicator
PART9	IOP	1	A	Terminal Yes-No response
PART9	IX	1	I	ISA + 1
PART9	SUB3	1	F	Total O&M cost—all bases
PART9	SUB1	1	F	Base total O&M cost
PART9	SUB2	1	F	Total facility O&M cost—one base
PART9	COST	1	F	Temporary location for various costs
PART9	RWX	1	F	Runway O&M costs
PART9	COST1	1	F	Civilian wages
PART9	COST2	1	F	Military pay and allowances
PART9	COST3	1	F	Aircraft fuel cost
PART9	ACOST	1	F	Aircraft O&M cost
PART9	BSUPP	1	F	Base support cost
PART9	IXX	1	I	New level of print detail input from terminal

* For variables in common, see PROGRAM PART2 variable dictionary, Table 18.

TABLE 46

PROGRAM PART9 PROGRAM DICTIONARY

PART9	Computes O&M costs and total systems costs

TABLE 47

PROGRAM PART9 LISTING

```
1009       C0MM0N IYEAR,ISWTCH(10)
1029       C0MM0N ACREQ(9,21),TBAS(9),TNAS(9),BPH(9,25),ASH(25,3),
1049      &ACFH(9,15),T0FF(9),TENL(9),TSTU(9),PNASE(9),SI(25),TCIV(9),
1069      &S0(25),FUREQ(9,3),PHPER(9,5),NBUSE(9),RW(25,3,3),
1089      &IACT(25,3),ACN01(25,3),T0FF1(25),EMT1(25)
1109       C0MM0N IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
1129      &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
1149      &RL0AD(21),C0MP(21),FLCST(21),A0M(21),CNAAC(21)
1169       C0MM0N NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),IBED(9),PEE(9),
1189      &PRE(9),P0(9),PS(9),PIE(9),TS(9),TH(9),TN0FF(9),TNENL(9),
1209      &TNCIV(9),ATCF(9),WR(9,2),TENAC(9,6),PERFAC(9),EMES(9)
1229       C0MM0N FAC0ST(50,6)
1249       C0MM0N FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),EXCH(10,2),
1269      &FAEM(8,2),TANKS(15),TAXIT0(3)
1289       C0MM0N IC0DES(50),IDES(50,3),RPI(50,9,2),IUNITS(50),
1309      &XRPI1(9,10,4),XRPI2(3,9)
1329       C0MM0N BR(50,9),XBR1(9,10,4),XBR2(3,9),DEF(50,9),
1349      &XDEF2(9),XDEF3(2,9),XDEF4(3,15,9),TEX(50,9),
1369      &NCAT,IYES,N0,IC0M,GT0TAL,NPH
1389       DIMENSI0N IARRAY(50)
1409       ALPHA IYES,N0,IC0M,I0P
1429       DATA IARRAY/1,27,28,2,3,4,5,6,7,29,30,8,9,10,26,11,12,13,
1449      &14,15,16,17,18,19,20,21,22,23,24,25,20*0/
1469       ISA=0
1489       IF(ISWTCH(9).EQ.3)ISA=1
1509       IF(ISWTCH(9).EQ.4)ISA=1
1529       IF(ISWTCH(9).NE.5)G0 T0 20
1549       PRINT,"D0 Y0U WANT DETAILED 0 & M C0STS (Y,N)"
1569    10 INPUT,I0P
1589       IF(I0P.EQ.IYES)G0 T0 17
1609       IF(I0P.EQ.N0)G0 T0 19
1629       PRINT,"INVALID REPLY--RETYPE"
1649       G0 T0 10
1669    17 ISA=2
1689       G0 T0 20
1709    19 ISA=1
1729    20 IX=ISA+1
1749       G0 T0 (35,25,30),IX
1769    25 PRINT 600
1789       G0 T0 35
1809    30 PRINT 601
1829    35 SUB3=0.
1849       D0 80 IB=1,9
```

TABLE 47 (Cont)

```
1869        SUB1=0.
1889        IF(NBUSE(IB).EQ.0)GØ TØ 80
1909        IF(ISA.EQ.2)PRINT 615,NASNAM(IB)
1929        SUB2=0.
1949        DØ 50 K=1,NCAT
1969        I=IARRAY(K)
1989        CØST=(RPI(I,IB,1)+RPI(I,IB,2))*FACØST(I,6)*1.407
2009        CØST=CØST/1000.
2029        IF(ISA.LT.2)GØ TØ 50
2049        IF(CØST.LT..5)GØ TØ 50
2069        PRINT 602,ICØDES(I),(IDES(I,J),J=1,3),CØST
2089   50   SUB2=SUB2+CØST
2109        RWX=0.
2129        DØ 60 K=1,10
2149   60   RWX=RWX+XRPI1(IB,K,2)
2169        RWX=RWX*200./9.
2189        CØST=RWX*.02*1.E-3
2209        IF(ISA.LT.2)GØ TØ 62
2229        PRINT 603,CØST
2249   62   SUB2=SUB2+CØST
2269        IF(ISA.EQ.2)PRINT 604,SUB2
2289        CØST1=PHPER(IB,5)*7.
2309        IF(ISA.EQ.2)PRINT 605,CØST1
2329        CØST=(TØFF(IB)-PHPER(IB,3))*15.771
2349        SUB1=SUB1+CØST
2369        IF(ISA.EQ.2)PRINT 606,CØST
2389        CØST=PHPER(IB,2)*5.849
2409        SUB1=SUB1+CØST
2429        IF(ISA.EQ.2)PRINT 607,CØST
2449        CØST=TSTU(IB)*8.411
2469        SUB1=SUB1+CØST
2489        IF(ISA.EQ.2)PRINT 608,CØST
2509        CØST=PHPER(IB,3)*13.911
2529        SUB1=SUB1+CØST
2549        IF(ISA.EQ.2)PRINT 609,CØST
2569        CØST=PHPER(IB,4)*5.849
2589        SUB1=SUB1+CØST
2609        IF(ISA.EQ.2)PRINT 610,CØST,SUB1
2629        CØST2=SUB1
2649        CØST3=(FUREQ(IB,1)*.127+FUREQ(IB,2)*.17+FUREQ(IB,3)*.17)
2669       &*1.E-3
2689        SUB1=SUB1+CØST3
2709        IF(ISA.EQ.2)PRINT 611,CØST3
```

TABLE 47 (Cont)

```
2729       AC0ST=0.
2749       D0 70 I=1,15
2769    70 AC0ST=AC0ST+ACREQ(IB,I)*ACFH(IB,I)*A0M(I)*1.E-3
2789       SUB1=SUB1+AÇ0ST
2809       IF(ISA.EQ.2)PRINT 612,AC0ST
2829       BSUPP=1246.13+1.42612*(TBAS(IB)-TNAS(IB))
2849       SUB1=SUB1+BSUPP
2869       IF(ISA.EQ.2)PRINT 613,BSUPP,SUB1
2889       IF(ISA.EQ.1)PRINT 614,NASNAM(IB),C0ST2,C0ST3,
2909      &AC0ST,BSUPP,SUB1
2929    80 SUB3=SUB3+SUB1
2949       IF(ISA.EQ.0)G0 T0 85
2969       PRINT 616,SUB3
2989    85 GT0TAL=GT0TAL+SUB3+6200.
3009       IF(ISA.EQ.0)G0 T0 90
3029       IF(ISA.NE.1)PRINT,"* N0N-ADD ITEMS"
3049    90 PRINT,"


                T0TAL SYSTEMS C0ST ="
3069       PRINT,"  FACILITY INVESTMENT C0STS"
3089       PRINT,"  + A/C INVESTMENT"
3109       PRINT,"  + 0 & M C0STS (LESS N0N ADD ITEMS)"
3129       PRINT,"  + CNATRA, CNABATRA, CNAVANTRA --- FIXED C0STS"
3149       PRINT 617,GT0TAL
3169       PRINT,"D0 Y0U WISH T0 RETURN T0 EXCESS-DEFICIENCY PR0GRAM (Y,
3189      &N)"
3209    92 INPUT,I0P
3229       IF(I0P.EQ.IYES)G0 T0 94
3249       IF(I0P.EQ.N0)G0 T0 95
3269       PRINT,"INVALID REPLY--RETYPE"
3289       G0 T0 92
3309    94 PRINT,"TYPE LEVEL 0F PRINTING DETAIL (1-5)"
3329       INPUT,IXX
3349       IF((IXX.LT.0).0R.(IXX.GT.5))G0 T0 94
3369       ISWTCH(9)=IXX
3389       ISWTCH(7)=1
3409       CHAIN "PART6*"
3429    95 PRINT,"D0 Y0U WISH T0 RUN F0R AN0THER YEAR (Y,N)"
3449   100 INPUT,I0P
3469       IF(I0P.EQ.IYES)G0 T0 120
3489       IF(I0P.EQ.N0)G0 T0 110
3509       PRINT,"INVALID REPLY--RETYPE"
```

TABLE 47 (Cont)

```
3529        GØ TØ 100
3549    110 STØP
3569    120 ISWTCH(10)=1
3589        IYEAR=IYEAR+1
3609        ISWTCH(6)=ISWTCH(6)+1
3629        ØPENFILE "RETURN"
3649        REWIND "RETURN"
3669        WRITE("RETURN",618)ICØDES,NBUSE
3689        WRITE("RETURN",619)IDES,IUNITS
3709        WRITE("RETURN",620)RPI,XRPI1,XRPI2,FACØST,BPH,CNAAC
3729        CLØSEFILE "RETURN"
3749        CHAIN "XLSRM*"
3769    600 FØRMAT(1X"SUMMARY Ø & M CØSTS"//1X"NAS    "
3789       &"MILITARY    A/C FUEL A/C Ø&M    BASE"/9X
3809       &"P&A"7X"TØTAL"4X"TØTAL"3X"SUPPØRT"3X"TØTAL")
3829    601 FØRMAT(1X"DETAILED Ø & M CØSTS (THØUS.)"/5X"FACILITIES"/
3849       &1X"CØDE  DESCRIPTIØN    CØST")
3869    602 FØRMAT(1XI5,1X3A4,1XF8.1)
3889    603 FØRMAT(1X"11110"1X"RUNWAYS"6XF8.1)
3909    604 FØRMAT(1X"*  SUBTØTAL"8XF8.1)
3929    605 FØRMAT(/1X"*  CIVILIAN WAGES"2XF8.1)
3949    606 FØRMAT(/1X"PAY & ALLØWANCES"/3X"PHASE ØFFICERS"3XF8.1)
3969    607 FØRMAT(3X"PHASE ENLISTED"3XF8.1)
3989    608 FØRMAT(3X"STUDENTS"9XF8.1)
4009    609 FØRMAT(3X"NAS ØFFICERS"5XF8.1)
4029    610 FØRMAT(3X"NAS ENLISTED"5XF8.1/4X"SUBTØTAL"8XF8.1)
4049    611 FØRMAT(/1X "A/C FUEL"11XF8.1)
4069    612 FØRMAT(1X"A/C Ø & M"10XF8.1)
4089    613 FØRMAT(1X"BASE SUPPØRT"7XF8.1/4X"TØTAL"11XF8.1/)
4109    614 FØRMAT(1XA4,2XF8.1,1X4(1XF8.1))
4129    615 FØRMAT(/25(1H-)/1X"NAS--"A4)
4149    616 FØRMAT(///4X"TØTAL Ø & M CØST"/6X"ALL BASES"5XF8.1)
4169    617 FØRMAT(1X9(1H-)10XF8.1)
4189    618 FØRMAT(8I8)
4209    619 FØRMAT(15A4)
4229    620 FØRMAT(5E13.6)
4249        END
```

XVII. UTILITY PROGRAMS

INTRODUCTION

17.1 The following four utility programs are associated with, but not a part of, the automated IFRS system: PROGRAMS TABGEN, ACFILIST, BAFILIST, and INFILIST. These programs support the IFRS system for they enable the user to rapidly modify data file TABLE* and to provide a list of the data in data files ACDAT*, BASED*, and INVCO*. This section briefly describes these utility programs and presents a program listing of each.

PROGRAM TABGEN

17.2 PROGRAM TABGEN is utilized to update the data file TABLE* which contains facility planning factors. Upon entry, the data currently contained in data file TABLE* are read into the computer's memory. Through a series of terminal responses, the user enters revisions to the planning factor data. When all revisions are complete, data file TABLE* is updated and the computer run is terminated. A program listing of TABGEN is shown in Table 48.

PROGRAMS ACFILIST, BAFILIST, AND INFILIST

17.3 PROGRAMS ACFILIST, BAFILIST, and INFILIST were developed to provide the user with an orderly list of the data contained in the data files ACDAT* (aircraft-oriented planning factors), BASED* (base-dependent planning factors), and INVCO* (facility cost data), respectively. When these programs are run, they read the appropriate data file and print the data with an appropriate descriptor. The computer runs are then terminated. Tables 49, 50, and 51 contain a list of PROGRAMS ACFILIST, BAFILIST, and INFILIST, respectively.

TABLE 48

PROGRAM TABGEN LISTING

```
100       DIMENSIØN FAPW(6),AP(4,3),GWTAB(3),FAMESS(7,2),
110      &EXCH(10,2),FAEM(8,2),TANKS(15),TAXITØ(3)
120       PRINT," TYPE 1 FØR GENERATE, 2 FØR UPDATE"
130       NSKIP=0
140       INPUT,ICHØICE
150       IF(ICHØICE.EQ.2)GØ TØ 500
160    10 PRINT," INPUT FAPW(6)"
170       PRINT," "
180       INPUT,(FAPW(I),I=1,6)
190       IF(ICHØICE.EQ.2)GØ TØ 505
200    20 PRINT," INPUT AP(4,3)"
210       PRINT," "
220       INPUT,((AP(I,J),J=1,3),I=1,4)
230       IF(ICHØICE.EQ.2)GØ TØ 505
240    30 PRINT," INPUT GWTAB(3)"
250       PRINT," "
260       INPUT,(GWTAB(I),I=1,3)
270       IF(ICHØICE.EQ.2)GØ TØ 505
280    40 PRINT," INPUT FAMESS(7,2)"
290       PRINT," "
300       INPUT,((FAMESS(I,J),J=1,2),I=1,7)
310       IF(ICHØICE.EQ.2)GØ TØ 505
320    50 PRINT," INPUT EXCH(10,2)"
330       PRINT," "
340       INPUT,((EXCH(I,J),J=1,2),I=1,10)
350       IF(ICHØICE.EQ.2)GØ TØ 505
360    60 PRINT," INPUT FAEM(8,2)"
370       PRINT," "
380       INPUT,((FAEM(I,J),J=1,2),I=1,8)
390       IF(ICHØICE.EQ.2)GØ TØ 505
400    70 PRINT," INPUT TANKS(15)"
410       PRINT," "
420       INPUT,(TANKS(I),I=1,15)
```

TABLE 48 (Cont)

```
      IF(ICHØICE.EQ.2)GØ TØ 505
   80 PRINT," INPUT TAXITØ(3)"
      PRINT," "
      INPUT,(TAXITØ(I),I=1,3)
      IF(ICHØICE.EQ.2)GØ TØ 505
      ØPENFILE "TABLE*"
  480 REWIND "TABLE*"
      WRITE("TABLE*",600)(FAPW(I1),I1=1,6),((AP(I2,J2),I2=1,4),
     &J2=1,3),(GWTAB(I3),I3=1,3),((FAMESS(I4,J4),I4=1,7),
     &J4=1,2),((EXCH(I5,J5),I5=1,10),J5=1,2),((FAEM(I6,J6),
     &I6=1,8),J6=1,2),(TANKS(I7),I7=1,15),(TAXITØ(I8),I8=1,3)
      PRINT," DØNE"
      STØP
  500 ØPENFILE "TABLE*"
      REWIND "TABLE*"
      READ("TABLE*",600)(FAPW(I1),I1=1,6),((AP(I2,J2),I2=1,4),
     &J2=1,3),(GWTAB(I3),I3=1,3),((FAMESS(I4,J4),I4=1,7),
     &J4=1,2),((EXCH(I5,J5),I5=1,10),J5=1,2),((FAEM(I6,J6),
     &I6=1,8),J6=1,2),(TANKS(I7),I7=1,15),(TAXITØ(I8),I8=1,3)
  505 PRINT," WHICH TABLE"
      IF(NSKIP.EQ.1)GØ TØ 510
      NSKIP=1
      PRINT," TYPE 1 FØR FAPW"
      PRINT,"      2 FØR AP"
      PRINT,"      3 FØR GWTAB"
      PRINT,"      4 FØR FAMESS"
      PRINT,"      5 FØR EXCH"
      PRINT,"      6 FØR FAEM"
      PRINT,"      7 FØR TANKS"
      PRINT,"      8 FØR TAXITØ"
      PRINT,"      0 FØR END ØF MØDIFICATIØNS"
  510 INPUT,M
      IF(M.EQ.0)GØ TØ 480
      GØ TØ (10,20,30,40,50,60,70,80),M
  600 FØRMAT(6E12.6)
      END
```

TABLE 49

PROGRAM ACFILIST LISTING

```
100       DIMENSIØN IATYPE(21),ACA(21),ACB(21),ACC(21),ACD(21),
110      &AHM(21),ACM(21),ASM1(21),ASM2(21),A(21,3),RNWYL(21),
120      &RLØAD(21),CØMP(21),FLCST(21),AØM(21),CNAAC(21)
130       ØPENFILE "ACDAT*"
140       REWIND "ACDAT*"
150       DØ 10 I=1,21
160       READ("ACDAT*",600)IATYPE(I)
170       READ("ACDAT*",601)LINE,ACA(I),ACB(I),ACC(I),ACD(I)
180       READ("ACDAT*",601)LINE,AHM(I),ACM(I),ASM1(I),ASM2(I)
190       READ("ACDAT*",601)LINE,(A(I,J),J=1,3)
200       READ("ACDAT*",601)LINE,RNWYL(I),RLØAD(I),CØMP(I)
210       READ("ACDAT*",601)LINE,FLCST(I),AØM(I)
220       READ("ACDAT*",601)LINE,CNAAC(I)
230    10 CØNTINUE
240       CLØSEFILE "ACDAT*"
250       DØ 30 J=1,3
260       ILØ=1
270       IHI=6
280       IF(J.EQ.2)ILØ=7
290       IF(J.EQ.2)IHI=10
300       IF(J.EQ.3)ILØ=16
310       IF(J.EQ.3)IHI=21
320       IF(J.EQ.3)PRINT 623,(IATYPE(I),I=16,21)
330       IF(J.EQ.3)GØ TØ 21
340       PRINT 602,(IATYPE(I),I=ILØ,IHI)
350    21 PRINT 603,(ACA(I),I=ILØ,IHI)
360       PRINT 604,(ACB(I),I=ILØ,IHI)
370       PRINT 605,(ACC(I),I=ILØ,IHI)
380       PRINT 606,(ACD(I),I=ILØ,IHI)
390       PRINT 607,(AHM(I),I=ILØ,IHI)
400       PRINT 608,(ACM(I),I=ILØ,IHI)
410       PRINT 609,(ASM1(I),I=ILØ,IHI)
420       PRINT 610,(ASM2(I),I=ILØ,IHI)
430       PRINT 611,(A(I,1),I=ILØ,IHI)
440       PRINT 612,(A(I,2),I=ILØ,IHI)
450       PRINT 613,(A(I,3),I=ILØ,IHI)
460       IF(J.EQ.3)GØ TØ 24
470       PRINT 614,(RNWYL(I),I=ILØ,IHI)
480    24 PRINT 615,(RLØAD(I),I=ILØ,IHI)
490       PRINT 616,(CØMP(I),I=ILØ,IHI)
```

TABLE 49 (Cont)

```
      IF(J.EQ.3)GØ TØ 40
      PRINT 617,(FLCST(I),I=ILØ,IHI)
      PRINT 618,(AØM(I),I=ILØ,IHI)
      PRINT 619,(CNAAC(I),I=ILØ,IHI)
   30 PRINT 620
   40 DØ 45 I=16,21
   45 FLCST(I)=FLCST(I)*1.E-3
      PRINT 621,(FLCST(I),I=16,21)
      PRINT 622,(AØM(I),I=16,21)
      PRINT 620
      STØP
  600 FØRMAT(5XA4)
  601 FØRMAT(V)
  602 FØRMAT(1X"TRAINING A/C"10X"UNIT "6(A4,3X))
  603 FØRMAT(1X"PARKING APRØN DATA A"4X"FT"6(F5.1,2X))
  604 FØRMAT(1X"PARKING APRØN DATA B"4X"FT"6(F5.1,2X))
  605 FØRMAT(1X"PARKING APRØN DATA C"4X"FT"6(F5.1,2X))
  606 FØRMAT(1X"PARKING APRØN DATA D"4X"FT"6(F5.1,2X))
  607 FØRMAT(1X"A/C PER HANGAR MØDULE"3X"AC "6(F4.0,3X))
  608 FØRMAT(1X"A/C PER CREW & EQUIP MDLAC "6(F4.0,3X))
  609 FØRMAT(1X"A/C PER BASIC SHØP MDL"2X"AC "6(F4.0,3X))
  610 FØRMAT(1X"A/C PER SUPPL SHØP MDL"2X"AC "6(F4.0,3X))
  611 FØRMAT(1X"CØVERED WAREHØUSE SPACE SF"6(F5.0,2X))
  612 FØRMAT(1X"SHED SPACE REQUIRED"5X"SF "6(F4.0,3X))
  613 FØRMAT(1X"ØPEN STØRAGE REQUIRED"3X"SF "6(F4.0,3X))
  614 FØRMAT(1X"RUNWAY LENGTH REQ."6X"LF"6(F6.0,1X))
  615 FØRMAT(1X"RUNWAY LØAD FACTØR"6X"**"3X6(F2.0,5X))
  616 FØRMAT(1X"RUNWAY CØMPØSITIØN FACT.**"3X6(F2.0,5X))
  617 FØRMAT(1X"INVESTMENT CØST (THØU.) $$"6(F5.0,2X))
  618 FØRMAT(1X"Ø&M CØST PER FLIGHT HØUR$$"6(F5.2,2X))
  619 FØRMAT(1X"INVENTØRY                UN"6(F5.0,2X))
  620 FØRMAT(///)
  621 FØRMAT(1X"ANNUAL FUEL (THØUS.)     GA"6(F5.0,2X))
  622 FØRMAT(1X"FUEL TYPE 1-JET 2=AGAS   **"6(F5.0,2X))
  623 FØRMAT(1X"TENANT A/C             UNIT "6(A4,3X))
      END
```

TABLE 50

PROGRAM BAFILIST LISTING

```
100       DIMENSIØN NASNAM(9),AD(9),PF(9,3),EL(9,3),CU(9),TH(9),TS(9),
110      &TNØFF(9),TNENL(9),TNCIV(9),PEE(9),PRE(9),PØ(9),PS(9),PIE(9),
120      &EMES(9),IBED(9),PERFAC(9),ATCF(9),WR(9,2),TENAC(9,6)
130       ØPENFILE "BASED*"
140       REWIND "BASED*"
150       DØ 10 I=1,9
160       READ("BASED*",600)NASNAM(I)
170       READ("BASED*",602)LINE,AD(I)
180       READ("BASED*",602)LINE,(PF(I,J),J=1,3),(EL(I,K),K=1,3)
190       READ("BASED*",602)LINE,CU(I),TH(I),TS(I)
200       READ("BASED*",602)LINE,TNØFF(I),TNENL(I),TNCIV(I)
210       READ("BASED*",602)LINE,PEE(I),PRE(I),PØ(I),PS(I),PIE(I)
220       READ("BASED*",602)LINE,EMES(I),IBED(I),PERFAC(I)
230       READ("BASED*",602)LINE,ATCF(I),(WR(I,J),J=1,2)
240       READ("BASED*",602)LINE,(TENAC(I,J),J=1,6)
250    10 CØNTINUE
260       CLØSEFILE "BASED*"
270       DØ 100 J=1,2
280       IL0=1
290       IF(J.EQ.2)IL0=9
300       IHI=8
310       IF(J.EQ.2)IHI=9
320       PRINT 601,(NASNAM(I),I=IL0,IHI)
330       PRINT 603,(AD(I),I=IL0,IHI)
340       PRINT 604,(PF(I,1),I=IL0,IHI)
350       PRINT 606,(PF(I,2),I=IL0,IHI)
360       PRINT 607,(PF(I,3),I=IL0,IHI)
370       PRINT 608,(EL(I,1),I=IL0,IHI)
380       PRINT 609,(EL(I,2),I=IL0,IHI)
390       PRINT 610,(EL(I,3),I=IL0,IHI)
400       PRINT 611,(CU(I),I=IL0,IHI)
410       PRINT 612,(TH(I),I=IL0,IHI)
420       PRINT 613,(TS(I),I=IL0,IHI)
430       PRINT 614,(TNØFF(I),I=IL0,IHI)
440       PRINT 615,(TNENL(I),I=IL0,IHI)
450       PRINT 616,(TNCIV(I),I=IL0,IHI)
460       PRINT 617,(PEE(I),I=IL0,IHI)
470       PRINT 618,(PRE(I),I=IL0,IHI)
480       PRINT 619,(PØ(I),I=IL0,IHI)
490       PRINT 620,(PS(I),I=IL0,IHI)
500       PRINT 621,(PIE(I),I=IL0,IHI)
510       PRINT 622,(EMES(I),I=IL0,IHI)
520       PRINT 623,(IBED(I),I=IL0,IHI)
530       PRINT 624,(PERFAC(I),I=IL0,IHI)
540       PRINT 634,(ATCF(I),I=IL0,IHI)
```

TABLE 50 (Cont)

```
      PRINT 625,(WR(I,1),I=IL0,IHI)
      PRINT 626,(WR(I,2),I=IL0,IHI)
      PRINT 627,(TENAC(I,1),I=IL0,IHI)
      PRINT 628,(TENAC(I,2),I=IL0,IHI)
      PRINT 629,(TENAC(I,3),I=IL0,IHI)
      PRINT 630,(TENAC(I,4),I=IL0,IHI)
      PRINT 631,(TENAC(I,5),I=IL0,IHI)
      PRINT 632,(TENAC(I,6),I=IL0,IHI)
  100 PRINT 633
      STØP
  600 FØRMAT(5XA4)
  601 FØRMAT(1X"NAS"17X8(1XA4,1X))
  602 FØRMAT(V)
  603 FØRMAT(1X"PARKING APRØN DEPTH "8(F5.0,1X))
  604 FØRMAT(1X"DAYS ØF READY FUEL STØRAGE:"/17X"JET  "8(F4.0,2X))
  606 FØRMAT(17X"AGAS "8(F4.0,2X))
  607 FØRMAT(17X"HEL0 "8(F4.0,2X))
  608 FØRMAT(1X"FUEL LØSS FACTØRS:"/17X"JET"3X8(F3.2,3X))
  609 FØRMAT(17X"AGAS"2X8(F3.2,3X))
  610 FØRMAT(17X"HEL0"2X8(F3.2,3X))
  611 FØRMAT(1X"ANNUAL CLASS UTILIZ."8(F6.0))
  612 FØRMAT(1X"TENANT DATA:"/4X"ANNUAL CLASS HRS"2X8(F4.0,2X))
  613 FØRMAT(11X"STUDENTS"2X8(F5.0,1X))
  614 FØRMAT(11X"ØFFICERS"2X8(F5.0,1X))
  615 FØRMAT(11X"ENLISTED"2X8(F5.0,1X))
  616 FØRMAT(11X"CIVILIAN"2X8(F5.0,1X))
  617 FØRMAT(1X"HØUSING DATA:"/2X"%ELLIGIBLE ENLISTED "8(F4.3,2X))
  618 FØRMAT(2X"%ENL. REQ. HØUSING  "8(F4.3,2X))
  619 FØRMAT(2X"%ØFF. REQ. HØUSING  "8(F4.3,2X))
  620 FØRMAT(2X"%STU. REQ. HØUSING  "8(F4.3,2X))
  621 FØRMAT(2X"%INELLIG.  ENLISTED "8(F4.3,2X))
  622 FØRMAT(1X"MESS HALL FACTØR"5X8(F3.2,3X))
  623 FØRMAT(1X"DISPENSARY 1=W/BEDS"3X8(I1,5X))
  624 FØRMAT(1X"%BASE REQ. ADMIN ØFF "8(F3.2,3X))
  625 FØRMAT(1X"WINDRØSE DATA MAIN"3X8(F3.2,3X))
  626 FØRMAT(10X"CRØSSWIND"3X8(F3.2,3X))
  627 FØRMAT(1X"TENANT A/C   VF"5X8(F4.0,2X))
  628 FØRMAT(14X"VT"5X8(F4.0,2X))
  629 FØRMAT(14X"VR"5X8(F4.0,2X))
  630 FØRMAT(14X"V0"5X8(F4.0,2X))
  631 FØRMAT(14X"VW"5X8(F4.0,2X))
  632 FØRMAT(14X"H"6X8(F4.0,2X))
  633 FØRMAT(///)
  634 FØRMAT(1X"ALT-TEMP CØRRECTIØN  "8(F3.2,3X))
      END
```

TABLE 51

PROGRAM INFILIST LISTING

```
100C-----THIS PRØGRAM READS CØST FILE & PRINTS EXISTING
110C-----VALUES IN TABLE IN FØRMATTED ØUTPUT
120       DIMENSIØN FACØST(50,6)
130       NCAT=30
140       ØPENFILE "INVCØ*"
150       REWIND "INVCØ*"
160       DØ 10 I=1,NCAT
170    10 READ("INVCØ*",600)LINE,(FACØST(I,J),J=1,6)
180       CLØSEFILE "INVCØ*"
190       PRINT 601
200       DØ 20 I=1,NCAT
210    20 PRINT 602,I,(FACØST(I,J),J=1,6)
220       STØP
230   600 FØRMAT(V)
240   601 FØRMAT(16X"FACILITY"8X"CØST-TIME"2X"CØST"/10X"UNIT"
250      &2X"SUPPØRT"1X"TYPICAL"1X"ADJUSTMENT"1X"TIME"1X"Ø&M"/
260      &1X"FACILITY"1X"CØST"2X"FACTØR"2X"SIZE"4X"FACTØR"5X"CØDE"
270      &1X"CØST/UNIT"/1X8(1H-)1X4(1H-)2X6(1H-)2X7(1H-)1X6(1H-)
275      &5X4(1H-)1X9(1H-))
280   602 FØRMAT(3XI2,2XF8.2,1XF4.2,3XF8.0,3XF4.2,6XF2.0,2XF6.2)
290       END
```